February 2018

Off-Grid Inverter with Regulated Output Voltage Amplitude

Niang Suan Thang
Worcester Polytechnic Institute

Ryan Joseph Cooney
Worcester Polytechnic Institute

Tianyi Xu
Worcester Polytechnic Institute

Off-Grid Inverter with Regulated Output Voltage Amplitude

A Major Qualifying Project

submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfilment of the requirements for the

degree of Bachelor of Science

by

Ryan Cooney

Niang Suan Thang

Tianyi Xu

Date:

March 2018

Report Submitted to:

Professor Shamsur Mazumder

Worcester Polytechnic Institute

Abstract

The paper discusses the design, simulation, and implementation of a 60W, $115V_{AC}$, 60Hz off-grid power inverter. Off-grid electric power is any power that is generated without connection to a conventional electric grid. An inverter converts a DC voltage to an AC voltage, which most household devices are compatible with. The DC to AC conversion stage of the constructed inverter is a low-voltage H-Bridge circuit. The H-Bridge circuit consists primarily of four transistors that are switched on and off in sequence to control the voltage across the load. The transistors are switched using 3-level pulse width modulation (PWM) generated by a microcontroller. PWM is a control strategy in which the duty cycle (pulse width) of a series of pulses is actively changed (modulated). 3-level PWM is unique in that it can generate three output voltages (V_{DC}, 0V, and $-V_{DC}$), whereas traditional 2-level PWM can only generate V_{DC} and $-V_{DC}$. After the H-Bridge, the circuit uses a low frequency step up transformer from 12V to mains voltage ($115V_{AC}$). As an improvement to previous inverter MQPs, a feedback system based on IQ sampling and a PID controller is implemented to maintain a constant output voltage amplitude over an input range of 10 to $15V_{DC}$. IQ sampling is commonly used in signal processing to determine the amplitude and phase of a sinusoid or combination of sinusoids. A PID (Proportional, Integral, and Derivative) controller is used to apply gain to the 3-level PWM switching scheme to correct the output voltage towards a setpoint. The inverter successfully powers small household loads such as a desktop fan and laptop. With a purely resistive load, the inverter has as low as 5.22% total harmonic distortion (THD), though may have over 17% THD depending on input voltage. The inverter output is distorted when an inductive or active load is powered. This motivates the need for a more elaborate output filter or active filtering in future projects.

Acknowledgements

We would like to express our sincere gratitude to several people without whom our project would not have been a success.

We would like to thank our MQP advisor, Professor Shamsur Mazumder, for his assistance, guidance and encouragement throughout our project. We are truly thankful for his constant support and willingness to dedicate so much of his time to our project.

We would like to thank Ian Costanzo for his guidance on designing our PCB, enabling us to successfully have a functioning first revision PCB.

We would like to thank Leah Morales for helping our team with soldering and the mechanical details of our project.

We would like to thank the WPI ECE department for the resources, funding, and lab space to complete our project.

Executive Summary

Off-grid solar power is projected to be used by nearly 100 million households worldwide by 2020 [1]. Several previous MQP projects have focused on developing off-grid $12V_{DC}$ to $115/120V_{RMS}$ inverters, which are a necessary component of any solar photovoltaic (PV) system used to power household devices. In most off-grid PV systems, the solar panel is used to charge a battery, and the inverter is used to allow standard $115/120V_{RMS}$ devices to be powered from the battery, as shown in Figure 1, which is a block diagram of our proposed system. The goal of this project was to build a 60W, pure sine wave inverter to fulfill this purpose. The goal of this project is unique for two reasons. First, at 60W, our inverter fills a market gap of very low power (less than 100W) pure sine wave inverters. Second, it integrates an output voltage regulation control system that was not implemented by previous inverter MQPs.

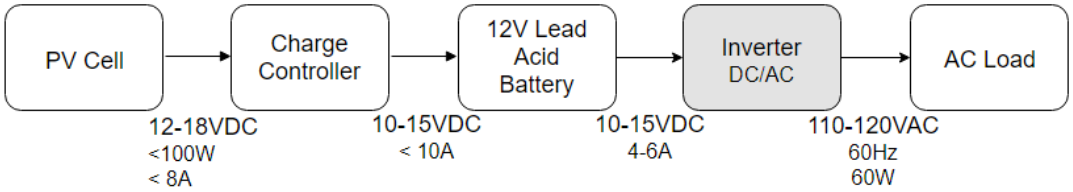


Figure 1: Off-grid solar block diagram

Several objectives must be completed to design an effective pure sine wave inverter. The total harmonic distortion (THD) must be low enough that the output is a clean sine wave that allows all sensitive electronics to run appropriately. The inverter must also convert energy as efficiently as possible. Additionally, the inverter must be able to maintain a stable output voltage and frequency regardless of the load and fluctuations in input voltage. Thus, our key design requirements, chosen primarily with the goal of matching the qualities of similar products on the market, were to design an inverter that could maintain an output voltage of $115V_{RMS} \pm 10\%$ at a frequency of 60Hz $\pm$ 0.1%, with greater than 80% efficiency and less than 4% THD. The inverter should be able to power small electronic loads, such as a laptop, small TV, desktop fan, etc., with no visible difference in function of the devices compared to how they function when powered by mains electricity.

A "Low Frequency Transformer" architecture was chosen for the inverter. This architecture, shown in Figure 2, uses a DC/AC converter at 12V followed by a step-up transformer, and was chosen primarily for its simplicity relative to other architectures.

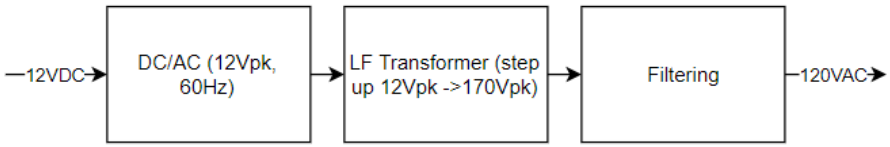


Figure 2: Low Frequency Inverter architecture

An N-channel MOSFET H-Bridge using 3-level pulse width modulation (PWM) was chosen for the DC/AC conversion stage. PWM is a control strategy in which the duty cycle (pulse width) of a series of pulses is actively changed (modulated). For our inverter,

the pulses are modulated by a sinusoid so that the average voltage of the pulse train is sinusoidal. 3-level PWM is unique in that it can generate three output voltages (V_{DC}, 0V, and $-V_{DC}$) to more closely approximate a sinusoid, whereas traditional 2-level PWM can only generate V_{DC} and $-V_{DC}$. The PWM is generated by a TI C2000 microcontroller, which we selected because it is designed specifically for power electronics and control systems. An LC lowpass filter with a cutoff frequency of 115Hz is used to smooth the 3-level PWM output into an approximate sinusoid. A feedback system, which was identified as a feature to improve the system over previous inverter MQPs, is used to sense the output voltage and regulate it by adjusting the PWM switching scheme. A unique method derived from IQ sampling was developed and is used internal to the microcontroller to calculate the amplitude of the output voltage.

The resulting inverter is shown in Figure 3. The main piece of the inverter is a custom-designed printed circuit board (PCB). The inverter also has a chassis mount transformer and inductor that are located off of the main PCB and are connected through terminal blocks, as well as an off-board switched mode power supply (SMPS) used to power the C2000 microcontroller. Under ideal circumstances (i.e. with an input voltage of 10V and 220Ω resistive load), the inverter was found to have a relatively pure sine wave output, shown in Figure 4 with a minimum of 5.22% THD. However, as input voltage increases close to 15V, THD increases above 17%.

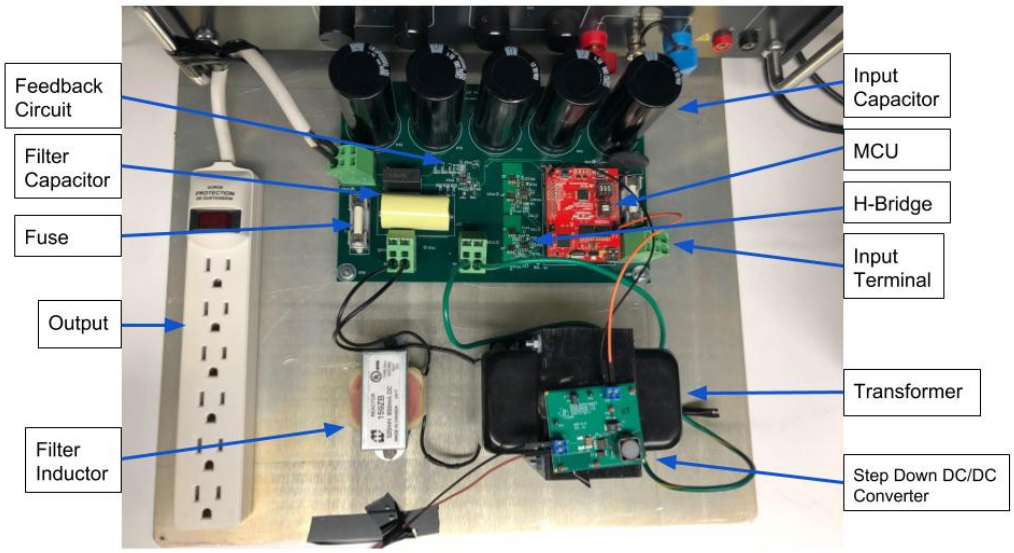


Figure 3: Completed inverter circuit

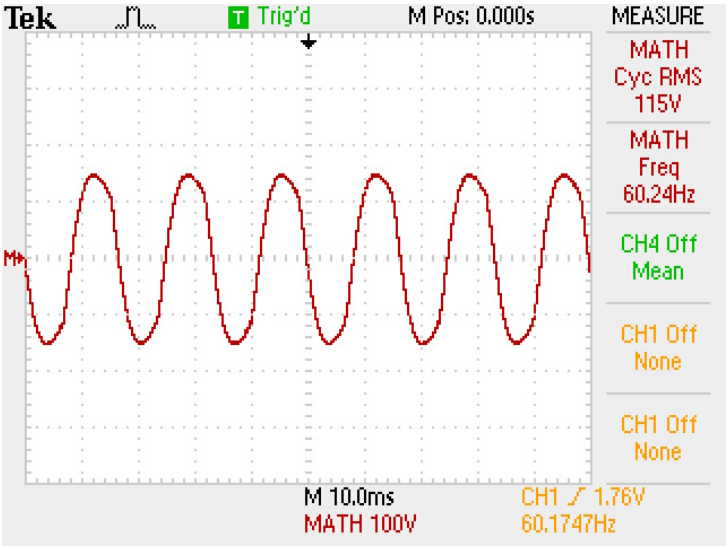


Figure 4: Output with 220Ω (60W) resistive load

The inverter is capable of powering a laptop adapter or desktop fan, although the output voltage will become visibly non-sinusoidal for any load that isn't purely resistive. The inverter was found to be 68 - 70% efficient under most conditions. The lack of efficiency is attributed mainly to the use of the large and inefficient 60Hz transformer, which dissipates over 21W when powering a 60W resistive load. Inefficiencies are also contributed primarily by the negative temperature coefficient (NTC) inrush current limiting resistor (5.7W) and output filter inductor (2.1W).

Several recommendations for future work follow from our results, two of which are discussed here. First, we would advise groups building similar devices against using the "Low Frequency Transformer" topology for an inverter. The low frequency transformer, though functional, proved to be large, heavy, expensive, and inefficient. It is worth the extra design effort to build (or buy) a DC/DC conversion stage for boosting voltage. Second, we recommend future student projects investigate the use of either (a) a more complex passive filter to allow the inverter to maintain low THD with various loads or (b) investigate the use of active filtering for the same purpose. The integration of active filtering into a feedback system similar to ours would be a natural extension of our work.

Authorship

Section	Ryan	Suan	Tianyi
Executive Summary	Contributed	-	-
1. Introduction	Contributed	Contributed	Contributed
2. Objective	Contributed	Contributed	Contributed
3. Background	3.3, 3.5, 3.6	3.1, 3.2, 3.4	3.7
4. Methodology	4.1, 4.2, 4.5, 4.6, 4.7, 4.8	4.3, 4.4, 4.11, 4.12	4.7, 4.9, 4.10
5. Simulation and Analysis	5.4, 5.5	5.3	5.1, 5.2
6. Implementation	6.2, 6.3	6.1	6.4
7. Results	7.6, 7.8	7.7	7.1, 7.2, 7.3, 7.4, 7.5
8. Conclusions and Recommendations	8.1	8.2	8.3

Contents

List of Figures

List of Tables

1 Introduction

Renewable energy is a growing market for several well-known reasons. First, renewable energy sources do not directly contribute to climate change through greenhouse gas emissions. Second, renewable energy sources can provide energy to places that are not reached by a utility grid. Third, renewable sources contribute to long term energy security because they are not dependent on finite fuel sources. Many renewable energy systems are in use today, with some of the most common being hydropower, wind power, and solar power. Solar energy is of particular interest for off-grid (non-utility tied) applications, because it is practical and cost effective to implement on a small scale, and makes use of an energy resource that is available everywhere on the planet.

Off-grid solar power is a market that is already booming and is projected to continue growing for the foreseeable future, as shown in Figure 5. By 2020, almost 100 million households worldwide are expected to use some sort of off-grid solar power.

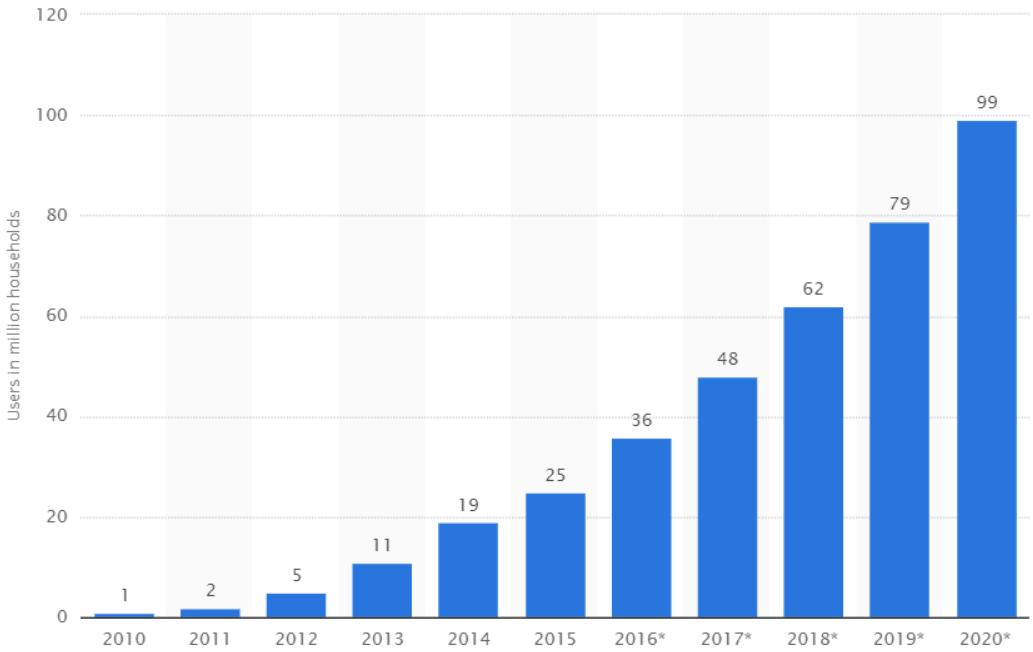


Figure 5: Projected global usage of off-grid solar power through 2020 [1]

Off grid solar can refer to any off-grid system that uses solar power in place of traditional utility electricity, and in which the photovoltaic (PV) system is not connected to a utility grid. This can refer to a variety of systems, from PV systems large enough to power a small community, to systems meant to power a single rural household, to small solar devices such as well pumps and lights. Our project will focus on developing an inverter for an off-grid PV system for a small household. An absolute minimalist summer home might use panels rated for as little as 100-300W, while a full-time off-grid home would be expected to use 1-3kW or more of solar panels [2]. Users of these homes and PV systems are likely to be powering typical rural residential devices, such as computers, TVs, washing machines, well pumps, etc.

A typical off-grid set-up is shown in Figure 6. In this system, a PV array creates a DC voltage, which may vary based on the instantaneous availability of solar energy.

The charge controller uses a DC/DC converter to control the voltage level reaching the battery bank. The charge controller is crucial because correctly changing the battery input voltage is crucial for ensuring a long battery life. The charge controller may also adjust its voltage and current draw from the solar panels to draw the maximum possible power from the panels. The battery bank (typically 12, 24, or 48V) supplies an inverter, which transforms the DC battery supply to a $115V_{RMS}$ or $120V_{RMS}$ AC voltage that can power household devices. Some inverters also function in reverse, converting an AC voltage from a generator to DC so that it can charge the batteries when solar power is unavailable.

It is worth noting that smaller off-grid set-ups will often use some devices powered directly by $12V_{DC}$, to avoid the cost and power losses of an inverter. However, as most devices that are commonly available are powered by $115/120V_{AC}$, inverters are nearly ubiquitous in off-grid systems.

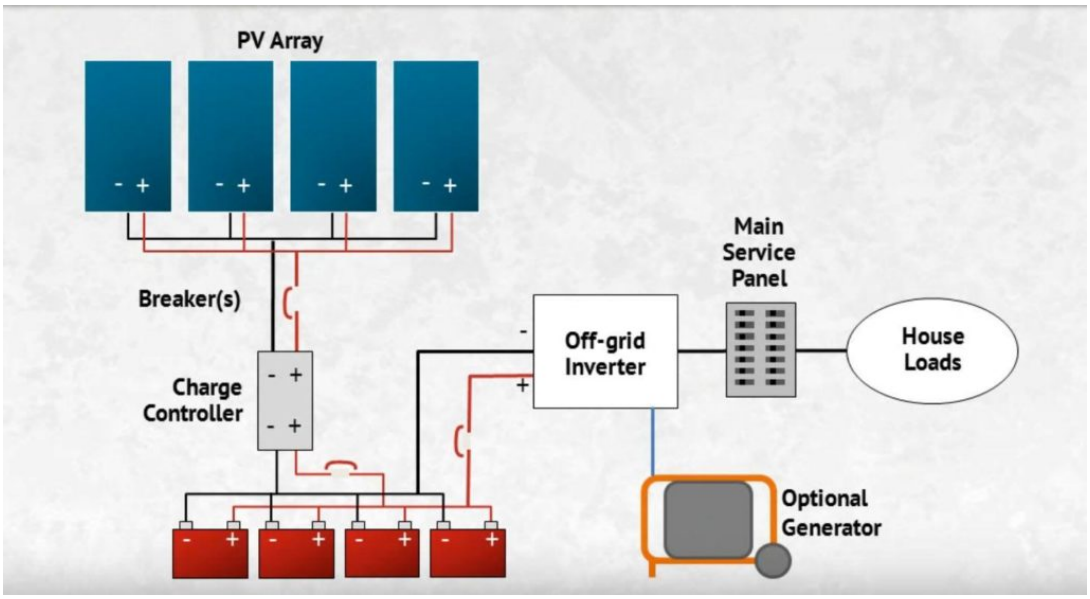


Figure 6: Off-grid solar block diagram [3]

Our project will focus on developing an inverter for an off-grid system that can power a load of up to 60W. Our solar inverter will fill a market gap of inverters marketed towards very small loads, on the scale of less than 100W. Typical inverters that are intended for solar applications do not run smaller than about 150W, though individual solar panels with a nominal power output of less than 100W are available. These panels would optimally be paired with similar sized and low priced inverters for users that only require power for small loads. Thus, our goal is to design and build a low power DC/AC inverter that is optimized for users of small loads, and can power a small load such as a laptop or desktop fan.

2 Objective

Problem Introduction

We will build a 3-level PWM inverter, and building on the work of previous MQP projects, will add a voltage control feedback system, which has not yet been implemented by an MQP team. We will ultimately implement the inverter circuit on a printed circuit board (PCB).

Project Objectives and Requirements

Our project goal is to research, simulate, design, and build a stand-alone inverter that will draw power from a 12V lead-acid battery and output up to 60W to power a small fan or laptop.

Objectives:

In order to reach this goal, we will attempt to meet the following objectives:

1. Evaluate inverter topologies based on efficiency, cost, novelty, and design risk.
2. Simulate our inverter design in National Instruments™ Multisim, and ensure that key specifications such as efficiency (>80%), output voltage (115V ± 10%), frequency (60Hz ± 0.1%) and THD (<4%) are met.
3. Design and build a PCB based on our simulation circuit, and use a microcontroller to implement closed loop feedback control of output voltage amplitude. The PCB circuit must meet all of the same requirements as simulation.
4. Verify the inverter PCB in a laboratory environment, then test the system's ability to power a 60W load with a 12V battery input.

Requirements

Based on our objectives and background research, we have determined customer and product requirements for our inverter. Customer requirements (Table 1) specify the expectations that a customer would have of our inverter, while product specifications (Table 3) detail the quantitative properties that will be required to meet those goals. We have divided the specifications and requirements into core requirements and additional requirements. The additional requirements (Tables 2 and 4) are those that may be important for a fully developed commercial product but that we have not attempted to complete. If the product were to go to market, in order for it to compete with currently available inverters, the additional features would be required as well. The background research informing many of these justifications is described in the following sections.

Customer Requirements–Core Requirements

Requirement	Justification and Details
Doesn't damage battery or loads or interfere with their operation (especially sensitive electronics).	Low cost "square wave" and "modified square wave" inverters interfere with the functioning of sensitive loads - many customers require a pure sine wave inverter.
Inverter can easily be configured to work with battery.	Different batteries may have different voltage output ranges, and the inverter should be compatible with various batteries.
Easy to install, plug and play.	For customer's ease of use.
Efficiency within range of inverters on the market - greater than 80%.	So that the inverter can be competitive in the marketplace. In off-grid systems, energy is at a premium, so efficiency is a key requirement.
Constantly able to power my 65W laptop.	This was our decision for a customer's need.
Won't burn or shock someone if they touch it.	As with all commercial products, must be safe to use.

Table 1: Customer requirements

Customer Requirements–Additional Requirements

Requirement	Justification and Details
Gives feedback on issues and performance (including amount of energy usage).	Common feature on most commercial inverters.
Doesn't waste power when no load is connected - less than 5 watts.	Similar commercial inverters specified a no-load power of less than about 5W.
Able to power the remote function on a TV, without wasting power.	Many larger inverters cannot power very small loads (such as the remote feature on a TV).
Complies to applicable regulations and has standard safety features.	Other inverters comply to UL 1741, IEC 62109 and other similar standards, and have several standard safety features.
Affordable - can be purchased for less than $150.	Similar commercial inverters are sold for less than $150.

Table 2: Additional requirements

Product Specifications–Core Specifications

Requirement	Justification and Details
Pure sine wave output, with THD $< 4\%$.	Other similar inverters specify THD $< 4\%$ or 5%. Test and verify with oscilloscope FFT function.
Efficiency of at least 80% at nominal power.	Same as other similar inverters (most are between 85-90%). Test and Verify by calculating P_{out}/P_{in}.
Nominal power output of at least 60W.	Meets our specified customer requirement of 60W. Test and verify by calculating power by equation $P = V * I$.
Output voltage 115VAC$\pm$10%.	Standard specification for mains voltage. Test and verify with oscilloscope.
Output frequency 60Hz$\pm$0.1%.	Standard specification for US frequency. Test and verify with oscilloscope.
Output current of at least 0.5A, nominal.	To meet power requirement at specified voltage.
Input range of 10V to 15V at $6A_{DC}$.	Same as similar inverters, matches the voltage range of 12V batteries and the current requirements to meet our load power requirement.

Table 3: Product specifications

Product Specifications–Additional Features

Requirement	Justification and Details
LED display, LCD screen or mobile/PC app to give customer feedback on current power consumption and inverter status.	Common feature on many commercial inverters.
Safety features: • Ground Fault Current Interrupt (GFCI) protection for outlets • Low voltage shutdown (output) • Overvoltage shutdown • Overtemperature shutdown • Short circuit shutdown	These are required for customer safety, and are the features that commercial inverters have.
Maximum price of approximately \$150.	Similar commercial inverters are sold for less than \$150.
Complies to UL 1741 and IEC 62109.	UL or IEC certification required for device to be sold commercially.
"Low battery shutdown" when input is less than 10V.	Lead acid batteries can be damaged if they are over-drained. Thus, 10V is a common specification for minimum working voltage for these batteries.

Table 4: Additional features and product specifications

System Block Diagram

We have developed the following high-level block diagram, shown in Figure 7, to describe our proposed high-level system. The inverter is highlighted in Figure 7 because it is the focus of this project.

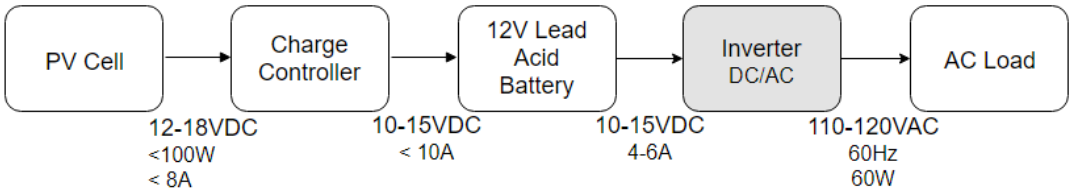


Figure 7: System block diagram

The PV cell will supply (nominal) $12V_{DC}$, though PV panels of this size often can output as high as 19V, and larger PV panels output higher voltages. The PV panel will be specified at 100W or less, though a larger panel is acceptable. This will be the input of the charge controller, which will use a DC/DC converter to supply our 12V battery. Because the battery voltage will vary between 10 and 15V, the charge controller's output will vary as well. The 12V battery output will then enter the inverter which will convert the battery voltage to $115V_{AC}$. Because the battery voltage can vary, the inverter will use feedback control to ensure a stable output voltage near 115V. This will power an AC load such as a laptop.

3 Background

3.1 Previous MQPs

Our MQP was inspired by a collection of past MQPs. The solar power and inverter related MQPs that have helped us develop the focus of our MQP are listed in the table below:

MQP Title	Year
An MPPT Charge Controller for Solar Powered Portable Devices	2017
An Exploration of Maximum Power Point Tracking Algorithms	2017
Smart Home Energy Controller	2017
Three Level PWM DC/AC Inverter using a Microcontroller	2012
PWM Techniques: A Pure Sine-Wave Inverter	2011
MPPT Charge Controller for Solar-Powered Portable Devices	2005

Table 5: Previous MQPs of relevance

While we referred to all of these MQPs in the process of deciding on our MQP topic, we discovered that two of these MQPs are most similar to our project, and our project is intended to build upon the work that they completed.

PWM Techniques: A Pure Sine Wave Inverter

This 2011-2012 MQP developed a pure sine wave inverter using 3-level pulse width modulation with an analog PWM generation circuit. The team used a DC/DC conversion stage to generate a 170V input. As part of the analog PWM circuit, a 60 Hz sine wave was generated with a "bubba oscillator" and compared with a high-frequency triangle wave to generate 2-level PWM. A 60 Hz square wave summing component was then used to create positive PWM for one half of the 60Hz cycle and negative PWM for the other half. The logic that resulted from the function generators controlled the two half-bridge MOSFET drivers, which in turn drove an H-Bridge to generate 3-level PWM. The H-Bridge was followed by a 4-pole LC filter. The team was successful in building a printed circuit board (PCB). Although the team set out with the intent of building a DC/DC boost stage and including a feedback system in the DC-DC conversion stage, they were unable to include the feedback system because they used a DC boost phase of an off-the-shelf inverter. Thus, they used a 170V input to their inverter, for a $120V_{RMS}$ AC output [4]. However, this MQP team did not use a microcontroller and feedback system in their project, which are the two main differences between our project and their project.

Three Level PWM DC/AC Inverter using a Microcontroller

This 2012-2013 MQP proposed an inverter design that converts DC to AC at an output power rating of 1kW using 3-level PWM. The same inverter topology (i.e. DC/DC boost stage followed by an H-Bridge) was used as in the previous MQP. However, the team used a microcontroller instead of analog control. The advantages of using a microcontroller for the control circuitry include that the microcontroller is used to achieve simpler voltage and frequency control, and that the microcontroller also supports an easier load feedback system. One of the team's recommendations was that, because the MSP430 microcontroller was not capable of sourcing enough current to directly driving the H-Bridge, it should be substituted for a different microcontroller, eliminating the

need for MOSFET drivers. This MQP also did not try adding a feedback system, which they suggested as additional future work for the project [5]. The main two differences between their projects and our projects are a) Our team used the TI C2000 microcontroller instead of the TI MSP430 b) Our team also built a feedback system.

3.2 Types of Photovoltaic System

Solar power for a home can be used as a sole source of power or in conjunction with other power sources. It is often used in conjunction with grid power (known as a grid-tied system), because power output from the sun is not constant and thus unreliable. However, with battery storage and smart power usage, solar power can provide all of the power needed without a grid connection. The differences between these two systems are explained below.

Grid-tied PV System

Homeowners who choose to set up a grid-tied PV system remain connected to the utility grid. By connection to the grid, the household is supplied with power at night when the solar panels do not generate electricity. Another advantage of an on-grid solar system is that the homeowner can sell any excess solar electricity back to the utility company [6]. The grid is said to act as a "virtual battery" for the system, because it absorbs excess electricity and can also supply electricity. As shown in Figure 8, the key elements of this system are the solar panels, inverter, and meter - the meter is used to track how much electricity is flowing into or out of the home. The inverter must synchronize to the frequency and phase of the utility grid in order to send power back to the grid. Additionally, the inverter must be able to quickly disconnect from the grid in the case of a grid fault. This system type is shown in Figure 8.

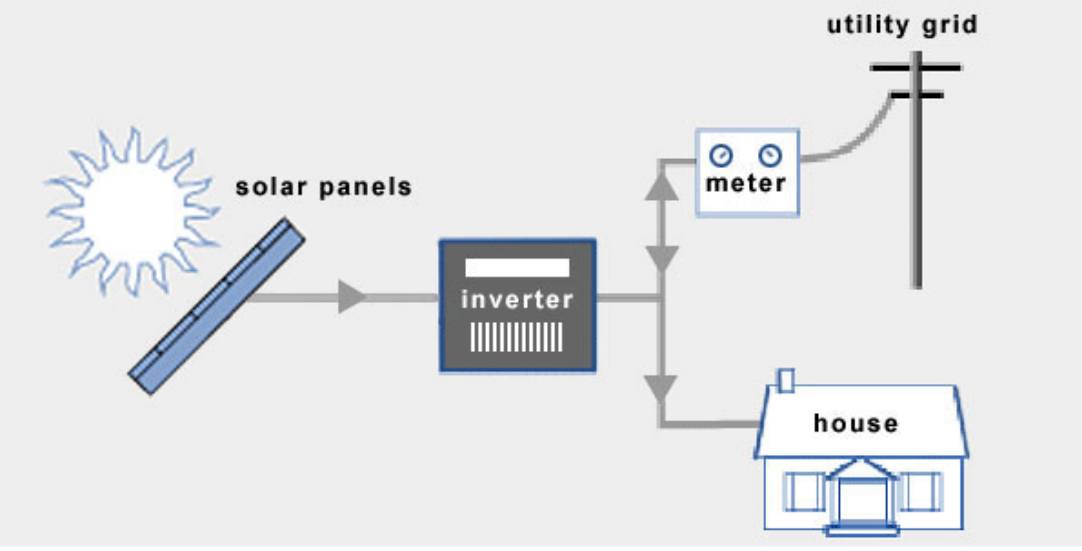


Figure 8: Grid-tied PV system [6]

Off-grid PV System

Off-grid PV systems depend entirely on their solar panels (and possibly other off-grid sources) and are not connected to the utility grid. Instead of the utility grid acting as a virtual battery, off grid systems have physical batteries which store energy from the solar

panels to be used when the sun is not producing power. When the sun is not providing energy, the batteries supply the inverter with stored energy which the inverter then turns into power for household appliances. A block diagram of this type of system is shown in Figure 9.

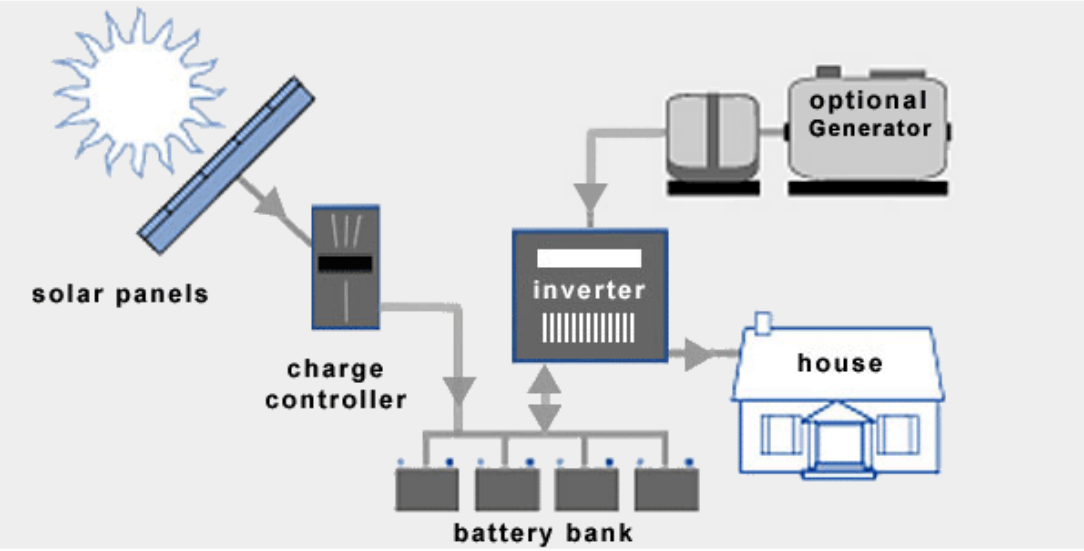


Figure 9: Off-grid PV system [6]

3.3 Inverter Topologies and Architectures

This section will cover several different aspects of an inverter topology. These include:

- Inverter output types (square wave, modified square wave, and pure sine wave)
- Inverter architectures; i.e. the relative location of the voltage boost and DC/AC stages within an inverter block diagram
- A description and comparison of PWM and multilevel inverter topologies and switching schemes

Inverter output types

Based on its load requirements, an off-grid inverter may have one of three output waveforms: (1) square wave, (2) modified square wave (also known as modified sine wave), or (3) pure sine wave. Each type of inverter output has different qualities in terms of efficiency, cost, complexity, and the devices that it can safely power. The three waveforms are shown in Figure 10.

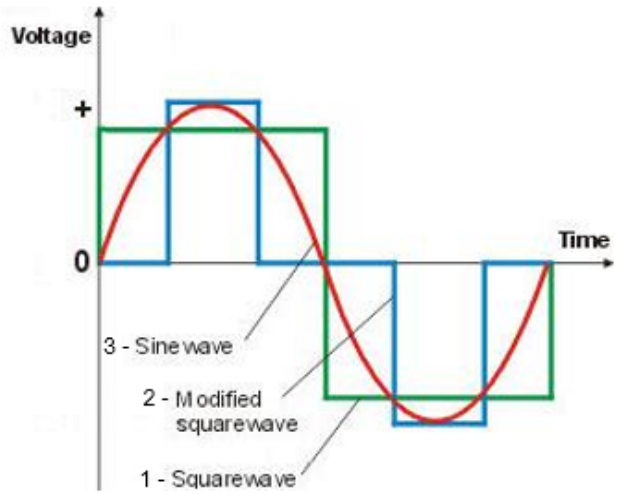


Figure 10: Different inverter output waveforms [7]

A square wave inverter produces a 60Hz square wave. Generally, only very small and cost-sensitive off-grid systems will use a square wave inverter. Loads that have an input AC/DC converter (i.e. DC loads) will generally be able to run on a square wave inverter with little issue, because rectification removes the low frequency harmonics that make a square wave inverter undesirable. Inductive loads generally will not work properly with a square wave inverter, because a square wave consists of steep voltage swings followed by constant voltage, which will cause excessive inductor currents and very poor power factor [8]. Some pros and cons of square wave inverters are listed in Table 6.

Pros	Cons
Lowest cost to design and purchase	Output voltage cannot be regulated
Simple and robust switching method - only need to switch each transistor twice per cycle	Large voltage THD - theoretically, 48.3% [9]
	May cause buzzing in certain devices due to harmonics at frequencies within human hearing range [10]
	Will cause devices that synchronize with 60Hz wave for timing to work incorrectly
	Less efficient, especially for inductive loads
	May damage equipment that requires a pure sine wave

Table 6: Pros and cons of square wave inverter

A modified square wave inverter output is essentially the same as a square wave, but it attempts to more closely replicate a sine wave by adding a voltage step at 0V. The pros and cons of a modified square wave topology are similar to those for a square wave topology, although voltage THD can be reduced and efficiency can thus be increased. The pros and cons of this topology are listed in Table 7.

Pros	Cons
Cheaper to buy than a sine wave inverter	Large voltage THD, though less than square wave.
Simple and robust switching method	May cause buzzing in certain devices due to harmonics at frequencies within human hearing range
Output voltage can be adjusted somewhat	Will cause devices that synchronize with 60Hz wave for timing to work incorrectly
	Less efficient, especially for inductive loads
	May damage equipment that requires a pure sine wave

Table 7: Pros and cons of modified square wave inverter

A pure sine wave inverter is meant to simulate the primarily sinusiodal voltage that devices connected to the grid will see. For sensitive devices or in most applications where high efficiency is crucial, a pure sine wave inverter is the clear winner. However, it has a more complicated design and potentially a need for more components (for more complicated switching techniques, output filtering, and voltage/current control), thus leading it to typically cost 2-3 times more than a comparable modified square wave inverter [11]. The low THD provided by a pure sine wave inverter is required for some sensitive (computer based) electronic devices to run correctly. For example, computer monitors or TV screens may show distorted pictures if THD is too high or if high frequency harmonics are present in the 60Hz wave. Also, motors may draw extra current (and thus produce extra heat) if certain harmonics produced by a square wave or modified square wave are too high. The pros and cons of a pure sine wave inverter are listed in Table 8.

Pros	Cons
Low voltage THD, typically less than 5%	Most expensive
All devices will run correctly	Most complicated to design, highest parts count
Efficiencies up to 98.8% [12]	
No buzzing sounds due to switching frequencies outside of human hearing range	
Voltage can be fully regulated	

Table 8: Pros and cons of pure sine wave inverter

Table 9 is a comparison of a modified square wave and sine wave inverter from the same series of products (both use a $12V_{DC}$ input). Note that the efficiency of the modified square wave inverter is not technically lower than that of the pure sine wave inverter, but that devices will typically run less efficiently with the modified square wave inverter. A square wave inverter is not included here because they are relatively rare given their similarities to modified square wave inverters but inferior performance.

Manufacturer	Name	Type	V_{out} (AC)	P_{out}	Efficiency	Voltage THD	Cost
Phocos	GP-SW150	Pure	110VAC ± 3%	150W	80-90%	< 3%	$147.99
Phocos	GP-175	Modified	110VAC ± 10%	175W	80-90%	(not given)	$25.90

Table 9: Comparison of modified square wave and pure sine wave inverters

Inverter Architectures

Different combinations of voltage boosting and DC/AC converting blocks can be used to design an inverter. Three architectures are common, which are (1) the Low Frequency Transformer design, (2) the High Frequency Transformer design, and (3) the Transformerless design.

A simple block diagram of the low frequency transformer architecture is shown in Figure 11. This architecture first converts $12V_{DC}$ to $12V_{PK\text{-}AC}$, before stepping the voltage up to $\sim 170V_{PK\text{-}AC}$ ($\sim 120V_{RMS\text{-}AC}$) through a low frequency transformer. In terms of parts count and design complexity this is the simplest topology. The low frequency transformer, however, is a large, heavy, and expensive component, and contributes significant inefficiencies. It has the advantage of built-in galvanic isolation at the output, protecting all internal circuitry. Galvanic isolation means that the input is "isolated" from the output such that if a short circuit occurs in the output side of the circuit, the input will be prevented from providing a large DC short circuit current. The low frequency transformer architecture is also advantageous in certain applications because of its ability to source significant inrush or surge current without the transformer saturating [13].

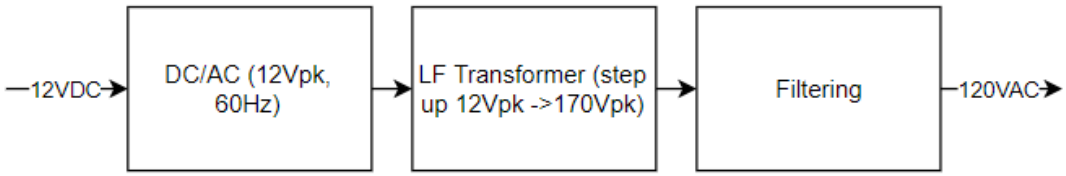


Figure 11: Low frequency transformer inverter architecture

The high frequency transformer architecture is shown in Figure 12. This architecture uses a DC/DC conversion stage. This stage steps up the voltage using both a DC/DC boost topology and a high frequency transformer. After being rectified, this voltage is DC/AC converted to $115/120V_{RMS\text{-}AC}$. This architecture is significantly more complicated than the low frequency transformer topology because of the circuitry required for the DC/DC converter, but does have numerous advantages. For one, high frequency transformers are smaller and have less losses than low frequency transformers. High frequency transformers are also cheaper, giving this architecture the possibility of being cheaper overall. Complicated design work and extra parts due to the DC/DC stage may actually increase cost relative to the low frequency transformer architecture, however. This architecture still has galvanic isolation, though this isolation does not protect the DC/AC conversion stage from the load. The high frequency transformer cannot support the same surge capacity as the low frequency variety [13].

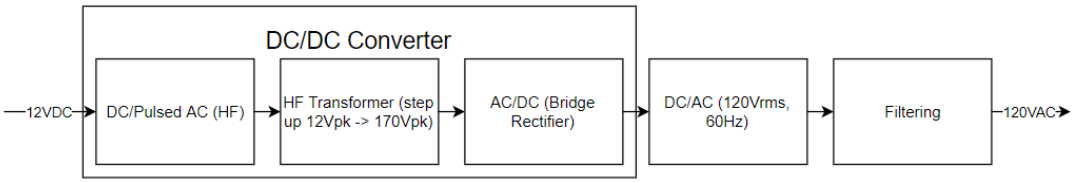


Figure 12: High frequency transformer inverter architecture

The transformerless inverter architecture is shown in Figure 13. Like the high frequency inverter, it uses a DC/DC stage to boost voltage. However, without having a transformer, this stage may be simplified, although large voltage step up/step down may be more difficult to realize without a transformer. It generally has the same advantages and disadvantages as the high frequency inverter. It also has additional advantages in efficiency, size, and cost of having no transformer whatsoever. However, with a large voltage step up system like ours (over 10x step up), a complicated DC/DC converter topology is necessary for efficient operation. Also, the lack of galvanic isolation has previously been in violation of inverter safety codes, and even though this topology is now permitted in the US, issues of grounding may limit the compatibility of this sort of inverter with some solar panel types. These inverters can be safely isolated with optical and capacitive isolation [13].

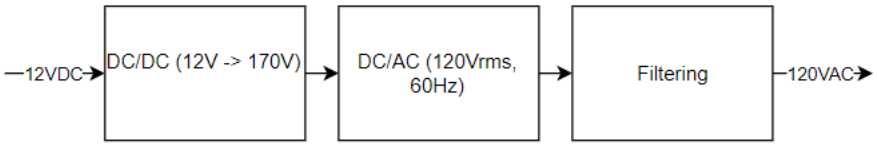


Figure 13: Transformerless inverter architecture

The performance of these three topologies is compared in several key areas in Table 10.

Field	Low Frequency Transformer	High Frequency Transformer	Transformerless
Efficiency	Typically less than 90%	Mid-high 90% range possible	1-2% more efficient than comparable HF transformer inverter for low voltage gains.
Complexity and Cost	Least complex; simple design can be cheapest but large magnetics raise cost	More complex; small transformer and detailed design raise cost	More complex; should be cheapest because it has least magnetics; detailed design may raise cost
Isolation	Built in, at output side	Built in, at input side	Must be designed for
Size	Largest	Medium	Smallest
Surge Capacity	Highest	Lower	Lower

Table 10: Comparison of inverter architectures [14], [15], [16]

DC/AC Converter Topologies: PWM

The 60Hz DC/AC converter within an inverter can be built in several different ways. Here, we discuss only topologies used to approximate a pure sine wave, although basic versions of these topologies are useful as square wave or modified square wave inverters.

The classic DC/AC converter topology is the Pulse Width Modulation (PWM) converter. PWM is a strategy for switching transistors in which each transistor is controlled with a rectangular wave that has fixed frequency and period but variable duty cycle. By adjusting the duty cycle, the average voltage of a load that receives current through a transistor can be controlled. For example, in Figure 14, three different PWM duty cycles are shown. In the case of a 0% duty cycle, the transistor being switched will never conduct; in the case of 100%, it will always conduct.

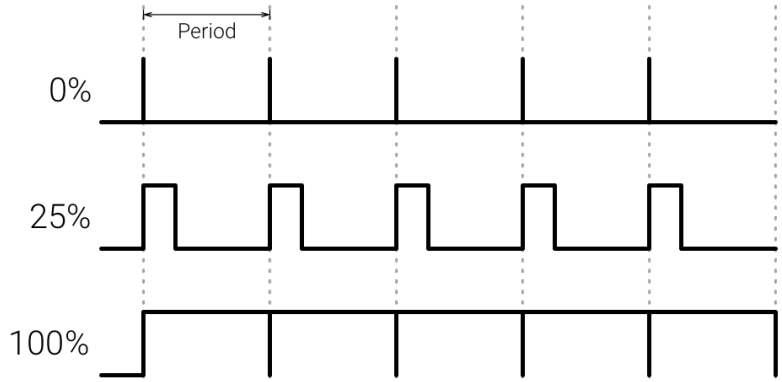


Figure 14: PWM with constant duty cycle [17]

One key note about PWM are the distinct harmonics generated as a result of varying duty cycles. A square wave (rectangular wave with 50% duty cycle) is well known for having only odd harmonics of its fundamental frequency. An example is shown in Figure 15, where the fundamental, 3rd, and 5th harmonics that are a part of the resulting square wave are shown.

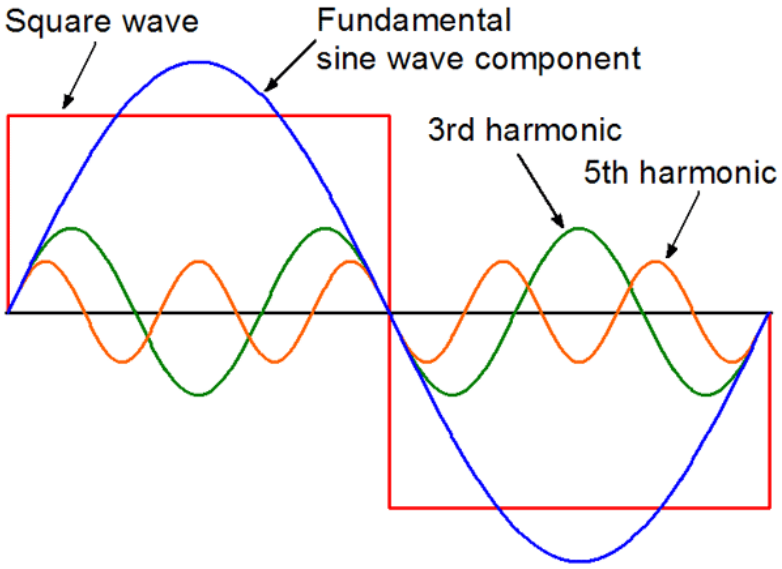


Figure 15: Square wave with fundamental frequency and 3rd and 5th harmonics [18]

The harmonic pattern for a square wave could also be restated as: every second harmonic is not present. This is a result of the 50% (1/2) duty cycle of the square wave. Thus, for a rectangular wave with a duty cycle of 25%, every 4th harmonic would be missing (25% = 1/4); for a wave with 20% duty cycle every 5th harmonic would be missing (20% = 1/5). This is important to consider because it means that lower duty cycles contribute greater numbers of harmonics. Waves with duty cycles of greater than 50% will contribute the same number of harmonics as their complementary duty cycles of less than 50%, for example, 20% and 80% duty cycle contribute the same number of harmonics.

The Fourier components of a square wave decrease as duty cycle increases. However, for a rectangular wave, higher frequency harmonics may have greater amplitudes than lower ones. Frequency spectra for a square and rectangular wave are shown in Figures 16 and 17 below with their corresponding waveforms. The rectangular wave has a duty cycle of just under 25%.

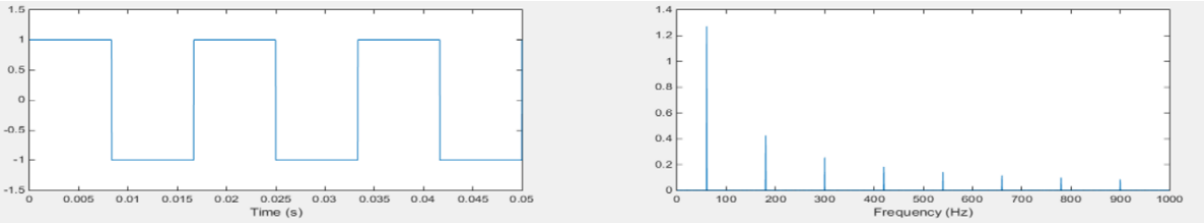


Figure 16: Square wave and corresponding FFT taken in MATLAB

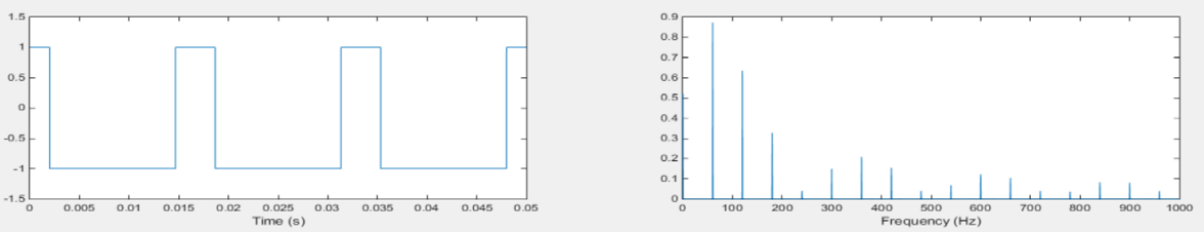


Figure 17: rectangular wave and corresponding FFT taken in MATLAB

In an inverter application, PWM is used to create a sine wave by changing the duty cycle of every pulse (unlike in the above discussion, in which duty cycle is constant). A typical PWM drive signal is shown Figure 18, and will be explained in more detail in the following sections. Note that for this PWM signal, although the duty cycle changes, the period between the middle of each pulse is approximately (though not exactly) constant. The average period of the PWM pulses is equal to that of the nominal switching frequency of the system.

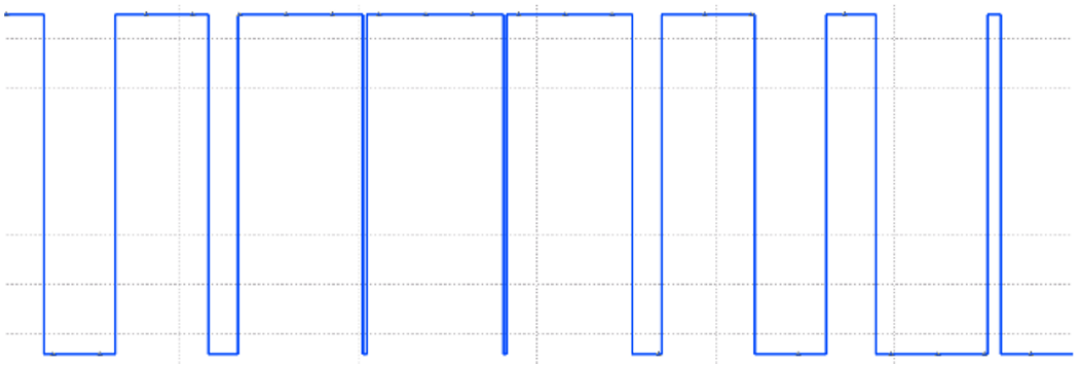


Figure 18: Simulated PWM pulse with varying duty cycle used to create a sine wave

For an inverter, PWM is often used to drive a circuit known as an H-Bridge. The operation of a simplified H-Bridge (with ideal switches in place of transistors) is shown in Figure 19. The converter operates by switching either side of the load to either V_{bat} or GND, thus allowing the load to experience a voltage drop of $+V_{bat}$, $-V_{bat}$, or 0V. 0V can be generated if both sides of the load are connected to V_{bat} (or if both sides are connected to GND).

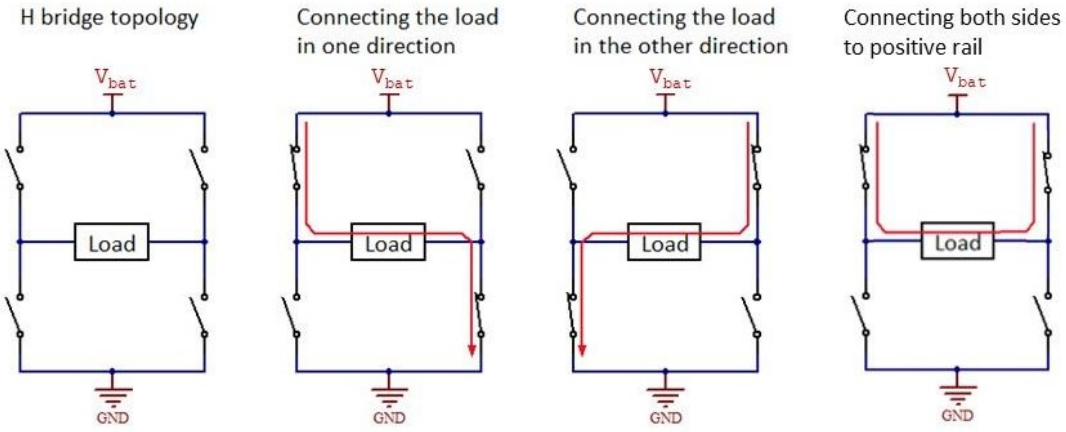


Figure 19: H-Bridge Switch Positions [19]

The goal of a PWM inverter is to closely approximate a sine wave by allowing the average voltage across the load to be sinusoidal. With additional filtering, this average voltage can become a very close approximation of a sine wave, typically with THD $<$ 4-5%. Two versions of PWM are shown below in Figures 20 and 21, and more detailed explanations will follow. These two PWM types are (respectively) 2-level and 3-level PWM. In both methods, the H-Bridge transistors are switched at "high frequency", i.e. at a much higher frequency than the fundamental frequency (often in the range of 10's to 100's of kHz). Thus, the output waveform consists of high frequency pulses.

Figure 20 is a 2-level PWM output, with corresponding 60Hz sine wave. In this PWM method, the voltage is alternated between $\pm V_{bat}$, where V_{bat} is the DC supply voltage [20].

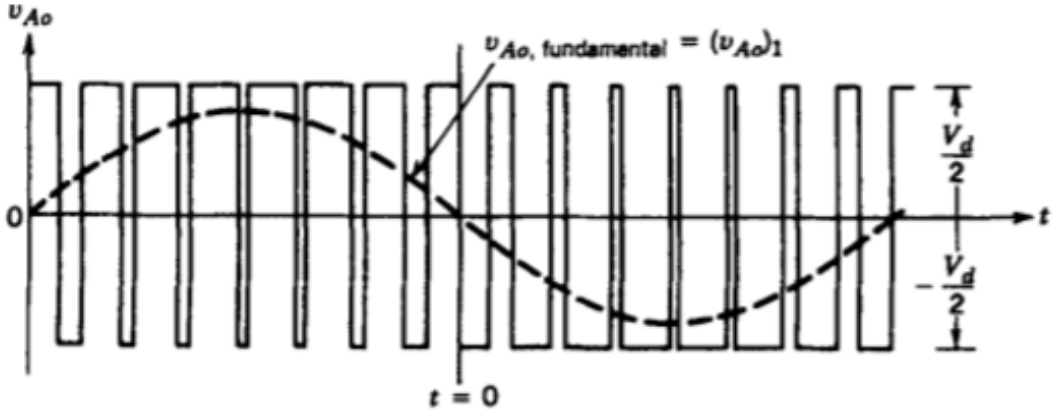


Figure 20: 2-level PWM [20]

Figure 21 is a 3-level PWM output. In this PWM method, the voltage is alternated between $+V_{bat}$ and 0V for half of the 60Hz cycle, then alternated between $-V_{bat}$ and 0V for the other half of the cycle.

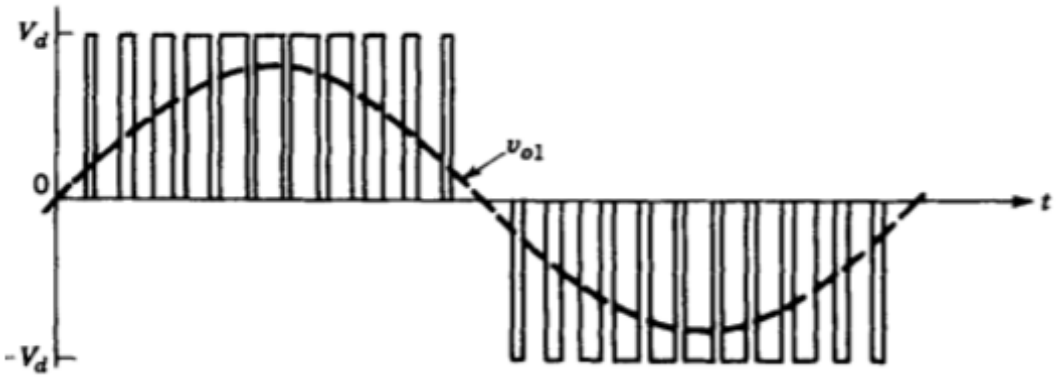


Figure 21: 3-level PWM [20]

2-Level PWM

In 2-level PWM, a high voltage of $+V_{bat}$ can be created by connecting the positive side of the load to the positive rail and the negative side of the load to ground. A low voltage of $-V_{bat}$ can be created by reversing which side of the load is connected to which rail. To achieve this (based on the H-Bridge shown in Figure 22), either switches S1 and S2 will be closed and S3 and S4 will be open, or vice versa [9].

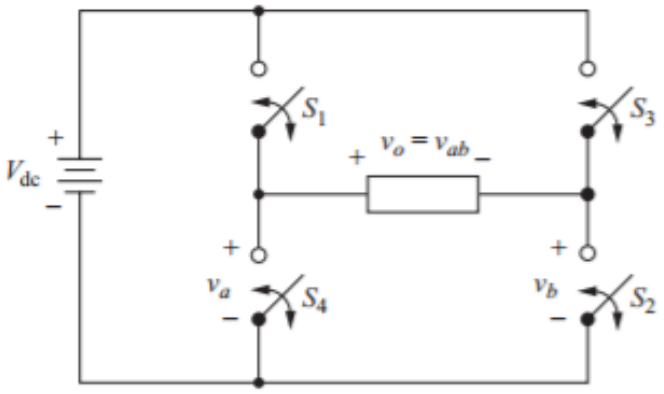


Figure 22: H-Bridge [9]

The gate drive for these switches is created by comparing a "reference" sine wave and a "carrier" wave, which is a triangle wave that is modulated at high frequency. Switch S1 and S2 are turned on when the $v_{sine} > v_{tri}$, and at the same time the other two switches are turned off. When $v_{tri} > v_{sine}$, S3 and S4 are turned on instead [9].

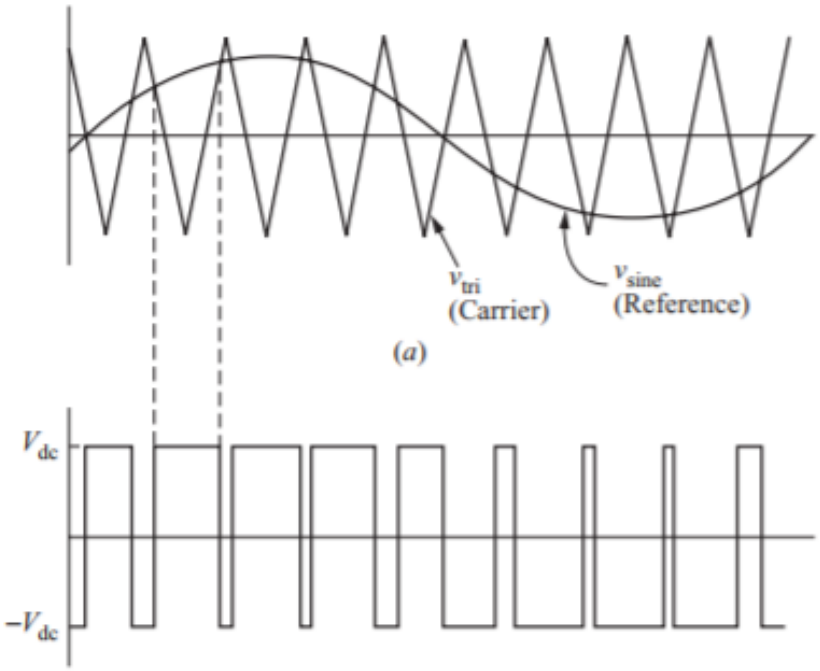


Figure 23: Generating 2-level PWM [9]

3-Level PWM

In 3-level PWM, shown in Figure 24, the output voltage can be either $+V_{bat}$, $-V_{bat}$, or 0V. There are two common switching methods used for 3-level PWM. The first method involves using two carrier sine waves that are 180° out of phase, and driving all four transistors at high frequency [9].

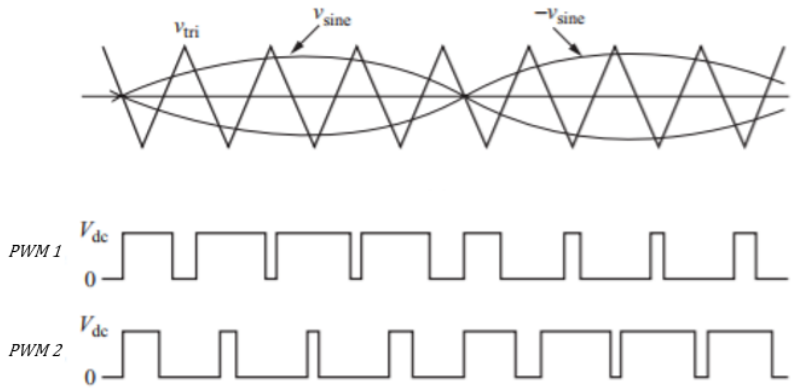


Figure 24: Generating 3-level PWM - first method [9]

Unlike 2-level PWM, where pairs of transistors are driven together, in 3-level PWM each transistor has its own unique drive signal. These are outlined below.

- S1 is on when $v_{sine} > v_{tri}$
- S2 is on when $-v_{sine} < v_{tri}$
- S3 is on when $-v_{sine} > v_{tri}$
- S4 is on when $v_{sine} < v_{tri}$

As a result, the voltage across the load is effectively PWM1 - PWM2 (as labeled in Figure 24). For the first half of the sine wave cycle, notice that PWM2 is only positive when PWM1 is as well. Thus, PWM1 - PWM2 is always either +VBAT or 0V for the first half of the cycle. For the second half of the cycle, PWM1 is only positive when PWM2 is as well. Thus, PWM1 - PWM2 is always either -VBAT or 0V for the second half of the cycle.

The second modulation strategy for 3-level PWM involves driving one pair of switches at high frequency and the second pair with a 60Hz (i.e. fundamental frequency) square wave [9]. The two drive signals are shown in Figure 25 - the "high frequency" signal is shown on top of the 60Hz square wave.

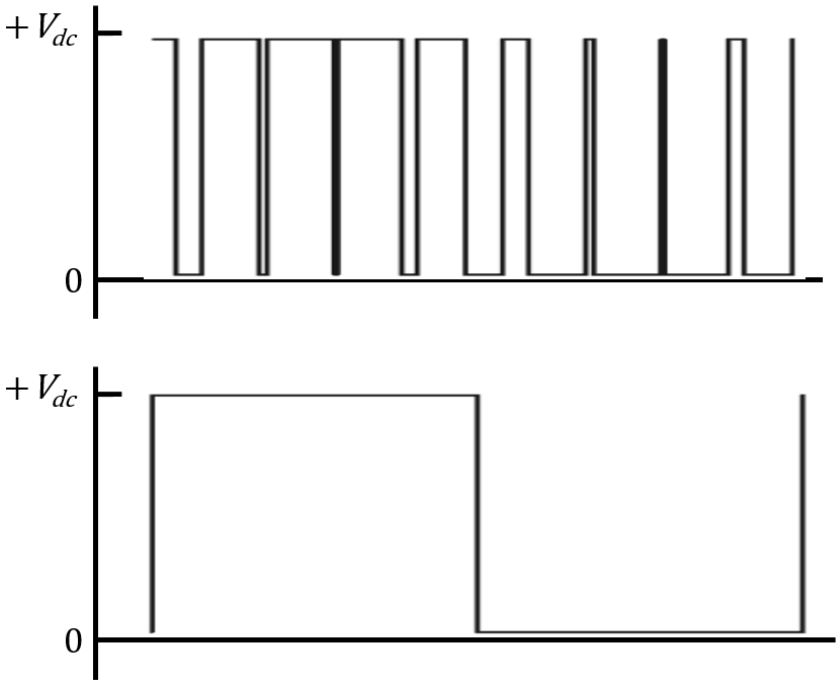


Figure 25: Generating 3-level PWM - second method

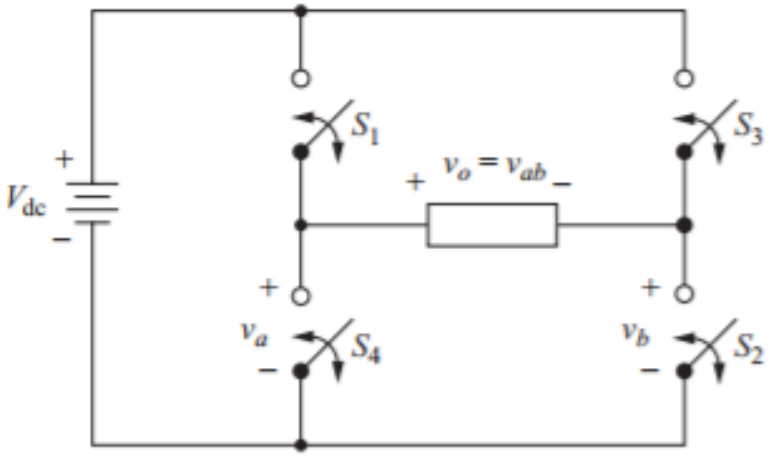


Figure 26: H-Bridge [9]

- S1 is on when $v_{sine} > v_{tri}$
- S2 is on when $v_{square} > 0$
- S3 is on when $v_{square} = 0$
- S4 is on when $v_{sine} < v_{tri}$

During the first half cycle, when the square wave voltage is high, S3 will short the negative side of the load to V_{bat} and S2 will be an open circuit. During this time, S1 and S4 (left pair) are alternatively switched at high frequency. When S1 conducts, both sides of the load are connected to high voltage, and thus the load voltage is 0V. When S4

conducts, the positive side of the load is grounded, and thus the net voltage across the load is -V_{bat}. When the square wave falls, S2 connects the low side of the load to ground and S3 is an open circuit. During this time, when S1 conducts, it connects the positive side of the load to V_{bat}, making the load voltage equal to +V_{bat}. When S4 conducts, both sides of the load are grounded, so its voltage is 0V [9]. Notice how in Figure 25 the widest and least-wide pulses in the high frequency wave align with the center points of the positive and zero portions of the square wave.

Comparison of 2 and 3-level PWM

The advantage of 2-level PWM is that it is relatively simple to control, because only two gate drive signals are needed, each controlling two transistors. However, it has greater THD than 3-level PWM at the same switching frequency because in 2-level PWM, the high frequency pulses have double the amplitude that the pulses have in 3-level PWM; the amplitude is 2*V_{bat} for 2-level vs V_{bat} for 3-level. When 3-level PWM is used with square wave voltage control, switching losses can also be reduced, because 2 switches operate at 60Hz instead of at high frequency.

Key Variables for PWM

1) Switching frequency

For PWM, the switching frequency is defined in terms of the frequency modulation ratio, m_f, defined as:

$$m_f = \frac{f_{tri}}{f_{sine}} \tag{1}$$

m_f is typically at least 21, although smaller values of m_f are possible. m_f is also typically an integer value, which is known as synchronous PWM. In synchronous PWM, harmonics due to switching are only in the range of f_{tri} and no low frequency harmonics near f_{sine} occur. In asynchronous PWM, where m_f is not an integer, subharmonics of the fundamental frequency can occur, but these harmonics are typically small at values of $m_f > 21$, and are acceptable for non-inductive loads [21].

2) Amplitude modulation ratio

PWM inverters allow the output voltage to be controlled by the amplitude modulation ratio, m_a, defined as:

$$m_a = \frac{V_{sine}}{V_{tri}} \tag{2}$$

If $m_a \leq 1$ (undermodulation), then V_{out} varies approximately linearly with m_a. Recall that because V_{out} is an AC voltage, we refer to its RMS value, and because V_{in} is a DC voltage, we refer to a constant value. Thus for $m_a \leq 1$, V_{out} will reach at most $V_{in}/\sqrt{2}$, which is the expected RMS value of an ideal sine wave.

The inverter can also be operated with $m_a > 1$ (overmodulation), but inverter output voltage will no longer vary linearly with input voltage in this region. Additionally, distortion and low frequency harmonics will increase with increasing m_a. Thus, as m_a increases past 1, output voltage will increase at a decreasing rate, until a point at which it simply becomes a square wave with an RMS voltage that is (ideally) identical to the input voltage [9].

3) Impact of m_a and m_f on harmonic content

A major advantage of a PWM technique is that the harmonics generated are at relatively high frequencies that are easily filtered, because they are far from the fundamental frequency. However, it is still worth noting that the harmonic content of a PWM generated inverter output will be significant before filtering. Additionally, the selection of m_a and m_f directly determine the frequencies and amplitudes of these harmonics. Harmonics are centered around the switching frequency, and theoretically, there should be no low frequency harmonics (i.e. the dreaded 3rd, 5th, 7th, etc. harmonics). The typical harmonic distribution of bipolar (2-level) and unipolar (3-level) PWM are shown in Figures 27 and 28. Note that harmonics are centered around all integer multiples of the switching frequency for bipolar, and around only even increments of the switching frequency for unipolar [9].

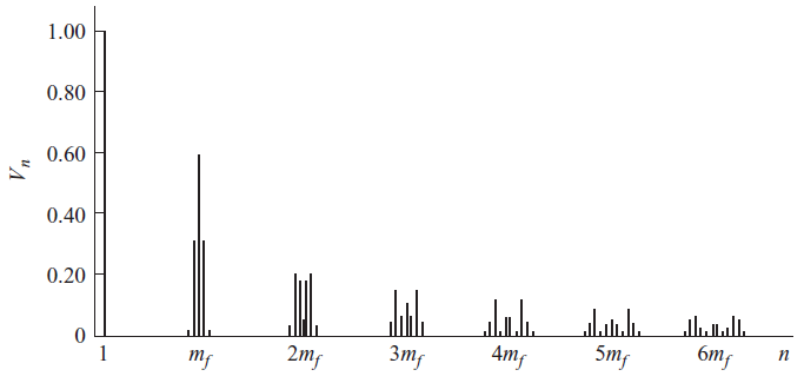


Figure 27: Harmonic content (in increments of fundamental frequency) for bipolar switching, with $m_a = 1$ [9]

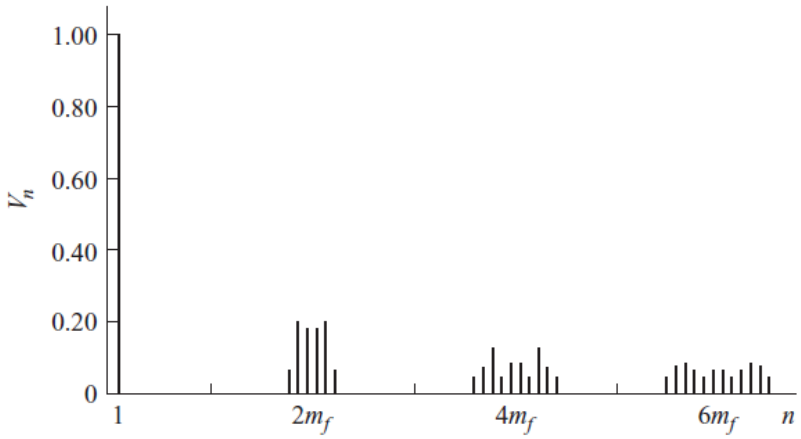


Figure 28: Harmonic content (in increments of fundamental frequency) for unipolar switching, with $m_a = 1$ [9]

If m_a is undermodulated, the frequencies at which harmonics occur will remain unchanged, but the amplitude of the harmonics will be changed, as shown in Table 29 for bipolar switching and Table 30 for unipolar switching. Note that reducing m_a will increase the amplitudes of high frequency harmonics relative to the fundamental frequency. Though this behavior is generally undesirable, the harmonics are all at high frequencies,

and thus can easily be filtered out. Recall then 'n' is the ratio of a given frequency to the fundamental - thus, with a fundamental of 60Hz, n = 1 represents 60Hz. m_f represents the ratio of the switching frequency to the fundamental, so n = m_f represents the harmonics at the switching frequency [9].

	m_a=1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
n=1	1.00	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
$n=m_f$	0.60	0.71	0.82	0.92	1.01	1.08	1.15	1.20	1.24	1.27
$n=mf\pm2$	0.32	0.27	0.22	0.17	0.13	0.09	0.06	0.03	0.02	0.00

Figure 29: Harmonic content for various values of m_a(V/V) for bipolar switching [9]

	m_a=1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
n=1	1.00	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
$n=2m_f\pm1$	0.18	0.25	0.31	0.35	0.37	0.36	0.33	0.27	0.19	0.10
$n=2m_f\pm3$	0.21	0.18	0.14	0.10	0.07	0.04	0.02	0.01	0.00	0.00

Figure 30: Harmonic content for various values of m_a(V/V) for unipolar switching [9]

4) Transistor selection

A third key component to H-Bridge operation is the selection of appropriate switching devices (typically, transistors). Several critical considerations for H-Bridge transistors are described here, though this is surely not an exhaustive list, especially for more complicated topologies and those experiencing very high/low power, high voltage, or high switching frequencies.

Because each transistor in an H-Bridge will conduct approximately 50% of the time, conducting losses can be a key element of inverter inefficiency. This is especially pertinent for H-Bridges such as ours that conduct relatively high currents. Thus, minimizing the $R_{DS\text{-}ON}$ value of an H-Bridge transistor is a design goal for transistor selection.

For transistors switching at high frequency, a non-negligible amount of power may be dissipated in driving the gates of the FETs. This is because the gate voltage of a FET is changed by charging the input gate capacitance (C_{ISS}), and current must flow through a small parasitic resistance (R_G) to charge C_{ISS}. The larger that each of these quantities becomes, the more current that it takes to bring the gate voltage to the required level in the same amount of time. This increased current in turn increases power loss due to driving the transistors. This is relevant for circuits that switch rapidly because (a) if a circuit is required to turn on/off quickly, a larger gate current (and thus power loss) is required to drive the switch, and (b) at higher frequencies, switching transitions happen more often and thus more power is dissipated on average.

The third area of FET losses is in switching. As the FET gate-source voltage increases, R_{DS} will decrease from $R_{DS\text{-}OFF}$ to $R_{DS\text{-}ON}$. As this occurs, current will begin to flow through the FET, due to the applied voltage across the FET and shrinking resistance from drain to source. Simultaneously, V_{DS}, which previously would have likely been large as $R_{DS\text{-}OFF}$ is quite large, would begin to drop. However, while V_{DS} is still fairly significant and while current is flowing through the FET, power losses will occur in the FET. Once R_{DS} reaches $R_{DS\text{-}ON}$, the transistor is fully conducting. This behavior is shown in simplified form in Figure 31.

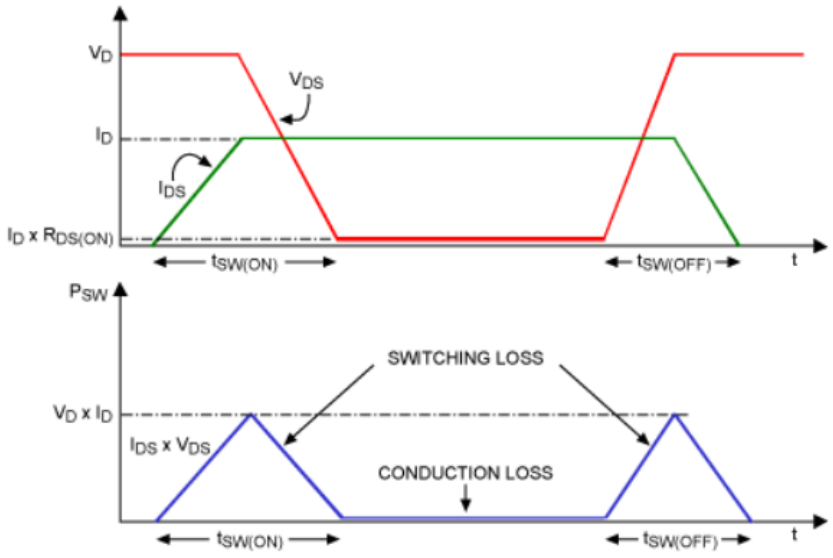


Figure 31: Switching losses [22]

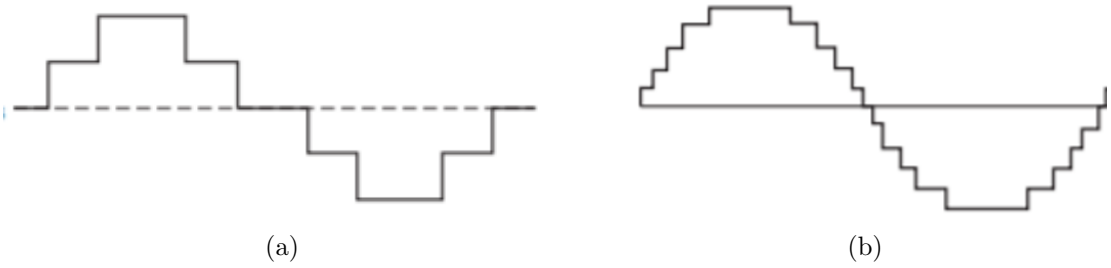
This can be mitigated by choosing a transistor that has a minimal turn-on and turn-off time, to minimize the amount of time spent switching. All aspects of transistors should also be balanced with the cost required to acquire them, because devices with low $R_{DS\text{-}ON}$, low turn-on and turn-off times, and low C_{ISS} and R_G will be more costly than a less ideal device.

5) Shoot-through voltage protection

The PWM strategies listed above do not account for the non-idealities of real switching devices, which take a non-zero amount of time to turn on. Because of this, if the low-side and high-side switches of either leg of an H-Bridge are switched simultaneously, there will be a moment where both transistors are conducting at the same time. This will result in a short circuit from V_{DC} to Ground, allowing large currents to flow and possibly destroying at least one transistor, or greatly shortening the lives of the switches. Because of this, practical PWM implementations need to allow a short time between turning off one switch in a leg and turning the other switch on. This is known as "deadtime" [23].

DC/AC Converter Topologies: Multilevel

Another category of inverter topology is known as the Multilevel inverter. Unlike a PWM inverter, a multilevel inverter uses low-frequency (at or near fundamental frequency) switching, although combinations of the two topologies are common and will be discussed. The operating principle of multilevel inverters is simple - create a "stepped" waveform that directly resembles a sine wave, rather than creating a signal whose average approximates a sine wave. As the name suggests, multilevel inverters have multiple levels, and the number of levels can vary. Figure 32 shows a 5-level multilevel waveform and 11-level waveform [9].



(a) (b)

Figure 32: (a) 5-level multilevel waveform and (b) 11-level multilevel waveform

Quite a few different multilevel inverter topologies exist. We make a distinction here between two types of multilevel topologies: topologies that use a single voltage source and topologies that require multiple isolated voltage sources. Due to the nature of our project, which uses a single PV panel and single battery, topologies involving multiple sources are not feasible, and will not be discussed. Rather, we will base our discussion on a single, relatively simple topology: the diode clamped multilevel topology (shown in Figure 33).

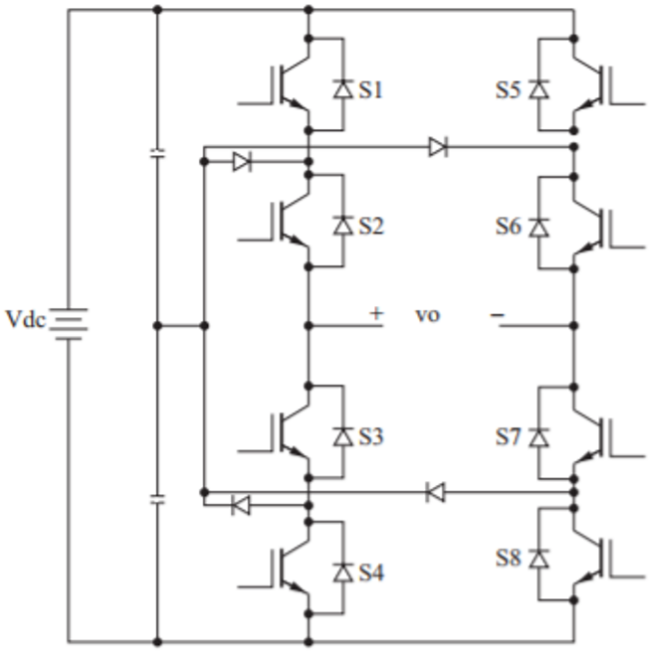


Figure 33: 5-level diode clamped inverter [9]

In this inverter, each leg (left and right) can be switched to deliver a voltage of either V_{DC}, 0V, or $\frac{1}{2}V_{DC}$, as shown in Figure 34 (only one leg of the inverter is shown here). The leftmost situation causes the load's positive end to be connected directly to V_{DC}; the middle situation connects the positive end of the load to ground, and the right situation connects the load to $\frac{1}{2}V_{DC}$, which is supplied by two identically valued capacitors connecting ground and V_{DC}.

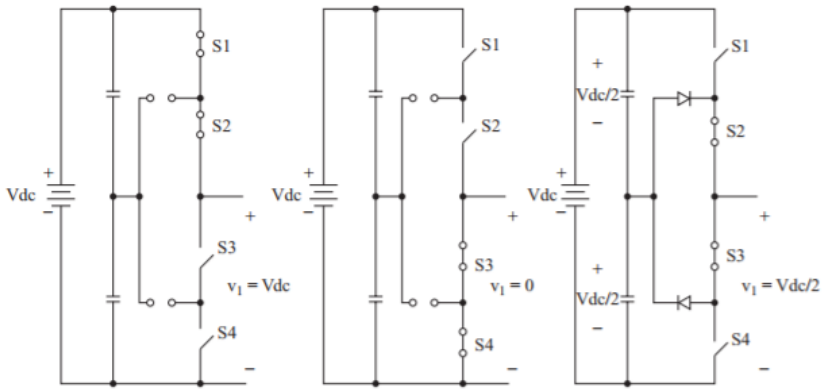


Figure 34: Operation of 5-level inverter [9]

Performance of the other side of the bridge is identical, but instead supplies voltage to the negative side of the load. The total load voltage is equal to the difference in positive and negative side load voltages, giving 5 possible voltage outputs for this circuit: V_{DC}, $\frac{1}{2}V_{DC}$, 0, $-\frac{1}{2}V_{DC}$, $-V_{DC}$. More levels can be added to this inverter by adding more transistor pairs.

The key application of this and other multilevel topologies is in high voltage systems. This is because this topology allows multiple transistors to share the battery voltage when they are turned off, as the transistors are in series between the load and rail voltages. This allows smaller, cheaper switching devices to be used for large systems [24].

Key Variables for Multilevel Inverters

1) Number of levels/switches

It is fairly obvious that a multilevel inverter with a greater number of steps will produce a cleaner sine wave with proportionally less THD. Interestingly, adding more levels (and thus more voltage transitions per cycle) will move THD to higher frequencies, making filtering simpler. However, the number of levels for a multilevel inverter is also directly proportional to the number of switching devices. While more switches will typically lead to a more expensive circuit, adding more switches allows the circuit to share high power/voltage between more devices, which may lower cost even while adding more components [9]. Thus, a key balance is between sine wave quality and cost.

2) Number of isolated voltage sources required

Many multilevel topologies require (or, take advantage of) the presence of multiple voltage sources. This property is ideal for solar farms and other resources that naturally have multiple isolated DC sources, and can consolidate distributed inverters into a single, high power, high quality sine wave inverter. This property is not ideal for systems such as ours that have only a single voltage source [9].

Extending PWM to Multilevel inverters

Some inverters apply PWM to multilevel topologies. Just as 3-level PWM reduced high frequency harmonics as compared to 2-level PWM, adding more levels can reduce the PWM harmonics of 3-level PWM and create a very high quality sine wave. Interestingly, one of the most efficient small scale inverters on the market right now uses this technology (also called "distributed switching"). The SolarEdge Power Optimizer and Inverter uses multilevel PWM to create an inverter that is up to 98.8% efficient. One example of such

an inverter output is shown in Figure 35.

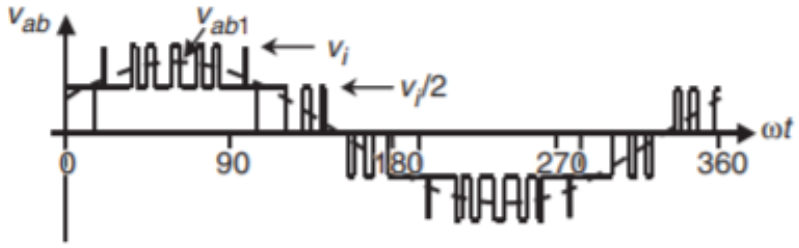


Figure 35: 5 level PWM [25]

3.4 Transistors

Types of Transistor

Transistors are used for amplifying signals or switching circuits. The three major types of transistors are Bipolar Junction Transistors (BJT), Field Effect Transistors (FET) and Insulated Gate Bipolar Transistor (IGBT). Their differences have been highlighted in Table 11.

	BJT	FET	IGBT
Method of Operation	Current Controlled	Voltage Controlled	Voltage Controlled
Input Impedence	Low	High	High
Swiching Speed	Slow (microsecond range)	Fast (nanosecond range)	Medium (between BJT and FET)
Voltage Rating	Low (< 1kV)	Low (< 1kV)	High (> 1kV)
Cost	Least Expensive	Medium	Most Expensive

Table 11: Comparison of transistor types [26]

For low power and low voltage applications such as ours, BJTs and FETs are most common because the high voltage capabilities of IGBTs are not needed. For a similarly sized BJT and FET, the BJT will switch faster. However, FETs can be manufactured at smaller sizes compared to BJTs, allowing FETs to have a smaller switching time than BJTs of similar capabilities. BJTs also tend to waste more power than FETs.

MOSFET as a switch

FETs have three modes of operation: triode, cut-off and saturation. In order for a FET to function as a switch, the transistor must operate in either the cut-off (effectively infinite resistance) or triode (small, pseudo-constant resistance) region. In the saturation region, a MOSFET acts as an amplifier, instead of a static resistance, as is desirable for a switch. For a MOSFET to be in cut-off, gate-source voltage must be less than the threshold voltage; i.e. $V_{GS}<V_{TH}$. For a MOSFET to operate in the triode region, two conditions must be satisfied. First, gate-source voltage must be greater than threshold voltage; i.e. $V_{GS}>V_{TH}$. Second, drain-source voltage must be smaller than the difference between gate-source and threshold voltage; i.e. $V_{DS}<V_{GS} - V_{TH}$.

GaNFET as a switch

One research section of our MQP was exploring a relatively new type of transistor: Gallium Nitride (GaN) FETs. Gallium Nitride transistors have only begun to be commercially available over the past 10 years or so, and a very limited selection of GaNFETs are available for purchase. Most of these GaNFETs are sold in die form, without a package, but a few packaged devices are available. As evidenced by their name, GaNFETs are field effect transistors like MOSFETs, and thus have the same basic relationships between gate, source, and drain voltage/current. GaNFETs are gaining popularity because of their extremely fast switching times and low on-resistances. For example, a study by Infineon investigating the performance advantage of using GaNFETs for an audio amplifier showed a dramatic improvement over MOSFETs, as shown in Figure 36.

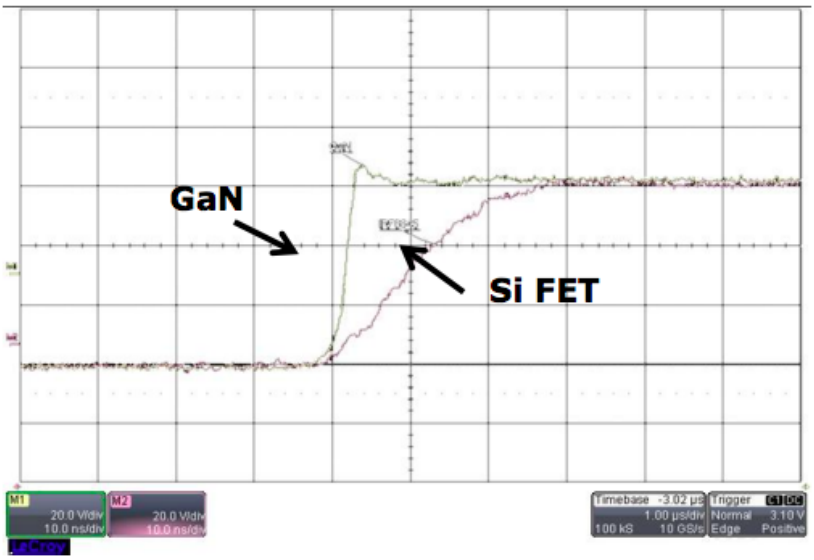


Figure 36: Turn on time of GaNFET and MOSFET [27]

Figure 37 shows the relationship between switching losses and FET current using several different varieties of GaNFET and MOSFET. The transistors, with voltage ratings ranging between 36V and 60V, were used in a 500kHz buck converter. GaNFETs showed much higher efficiencies primarily because of their lower switching losses, which become more significant as switching frequency increases [28].

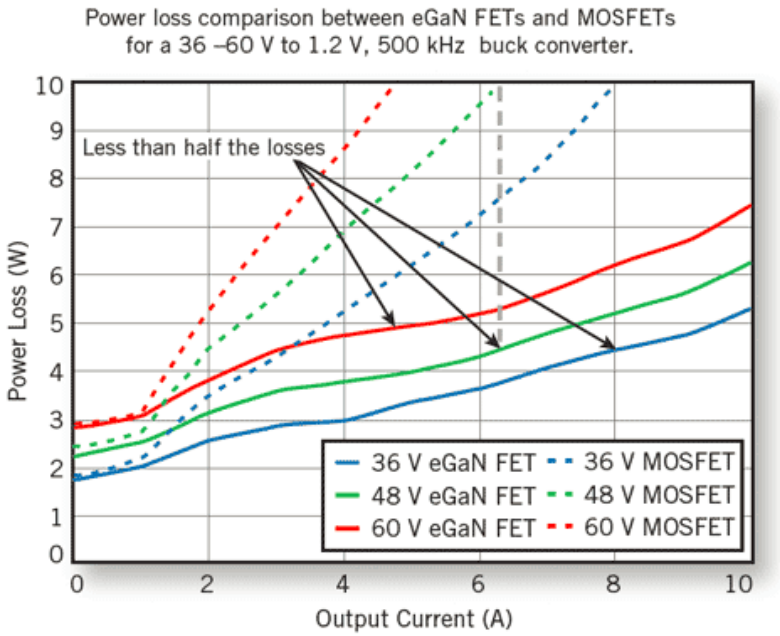


Figure 37: Power loss vs output current [28]

Table 12 is a product comparison table between a similarly rated MOSFET and GaNFET - both are N-Channel devices. Note the nearly identical voltage and current ratings of the two devices. All specifications were taken with a junction temperature of 25°C. Note that these devices were chosen from among pre-packaged devices available to purchase online.

	MOSFET	GaNFET
Manufacturer	Nexperia (NXP)	GaN Systems
Part Number	PSMN020-100YS	GS61004B
Drain to Source Voltage	100V	100V
Continuous Drain Current	43A	45A
Typical Drain-Source On-Resistance at high V_{GS}	15mΩ (V_{GS}= 10V)	10mΩ (V_{GS} = 6V)
Gate Charge	57.4nC	6.2nC
Turn on Time	26.1ns	5ns
Turn off Time	56.7ns	5ns
Input Capacitance	2980pF	328pF
Output Capacitance	226pF	133pF
Cost	1- $0.386	1- $5.64

Table 12: Comparison between similar MOSFET and GaNFET

3.5 Output Regulation for Inverters

Most inverter systems have control systems to maintain a stable output despite variable loads or input voltages. These control systems can be used to control output voltage and/or current by varying the PWM signals sent to the inverter. A relatively basic, single phase control loop used for output voltage amplitude regulation is shown in Figure 38.

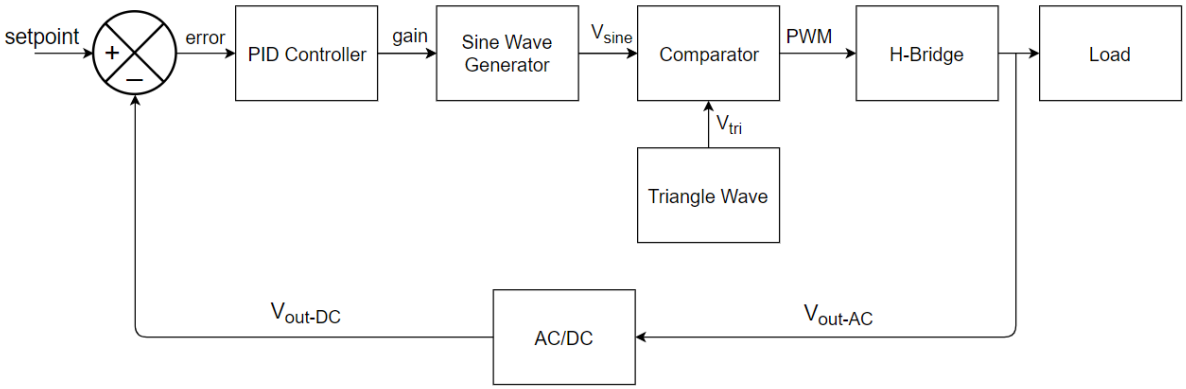


Figure 38: Feedback control loop (adapted from [25])

In Figure 38, the "setpoint" is a DC reference voltage - this control loop aims to hold the output voltage at the setpoint. The instantaneous output voltage ($V_{out\text{-}AC}$) is conditioned in some manner to create a representative DC value ($V_{out\text{-}DC}$), which is compared to the setpoint. The difference of these two voltages, the "error", is passed to a PID (Proportional, Integral, and Derivative) controller, which will be explained in the following section. The PID controller produces a scaled error signal known as the "gain". This signal is then used as a seed for generating PWM proportional to itself. In Figure 38, the gain is used to scale the PWM generator sine wave (recall that output voltage scales with V_{sine}). A PWM signal is generated by comparing V_{sine} and V_{tri}, which is used to drive the H-Bridge. The filtered output sine wave from the H-Bridge, which is used to power the load, is then fed back to be compared again. An important note here is that, although they are not shown in this diagram, the battery voltage feeding the H-Bridge and the load current draw can both vary with time, which may in turn impact $V_{out\text{-}AC}$.

PID stands for "Proportional, Integral, Derivative". Thus, a PID controller outputs the sum of 3 values: a proportional gain of the instantaneous value (K_P), a gain of the integral of the input (from t = 0 until the current time) (K_I), and a gain of the instantaneous derivative of the value (K_D). The equation for the output voltage from a PID is given by:

$$v(t) = K_P * e(t) + K_i * \int_0^t e(\tau) * d\tau + K_d * \frac{de}{dt} \tag{3}$$

The proportional term is used to amplify the current error so that it can be corrected; however, if only a proportional gain is used, there will be a constant, steady state error, as shown in Figure 39. In this image, the blue line is the setpoint (changing with time as a step function), and the other 4 lines represent output values with different gain levels. A larger proportional gain reduces steady state error, but note that too large of a proportional gain contributes to overshoot [29]. Also note that in this figure an additional stimulus was provided to the system just after t = 2s, which accounts for the ripple shown in the output values. Note that this figure and other figures in this section have "Temperature" on their y-axis; this is because the PID examples from [29] relate to a temperature control system.

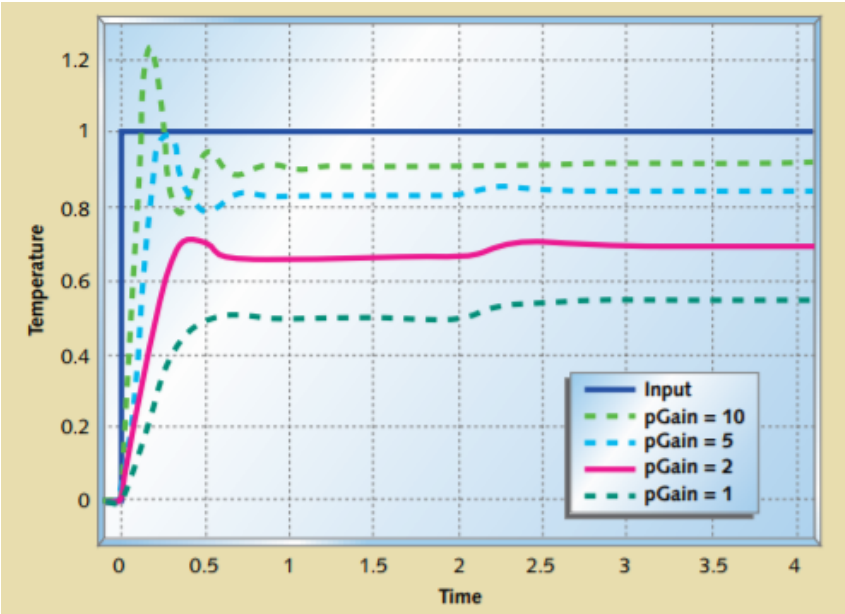
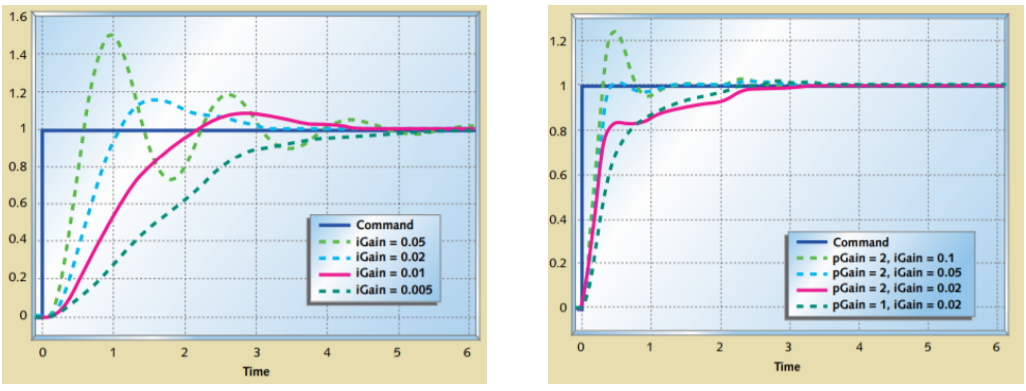


Figure 39: Impact of proportional gain [29]

The integral term serves two purposes. First, it corrects the steady state error, and second, if a proportional gain is not delivering a strong enough change in output voltage, the integral term will increase the gain with time and achieve the desired response. An integral term can also be used without a proportional term, as shown in Figure 40(a), and can eliminate steady state error. However, it is clear that in the figure, all of the various values of K_i either cause overshoot or have a much slower response than the purely proportional gain in Figure 39.

A PI controller output (with P and I gains) is shown in Figure 40(b). It is evident from this figure that the combination of P and I terms can allow the controller to reach a steady state without error and without the extremely slow response time of an integral-only controller. If properly "tuned", a PI controller can give a steady and fast responding output [29].



(a) (b)

Figure 40: (a) Impact of integral gain and (b) Impact of integral and proportional gain

The optional D term is used to prevent overshoot and counteract fast changes in the error term. K_d*Δerror is subtracted from the gain, so that when the change in error signal is large (leading K_p to produce overshoot), the response will slow itself. The derivative

term can contribute to instability, because small but fast changes in the error signal will then produce large derivative response that will contribute to further error [29]. The impact of adding the derivative term to the controller is shown in Figure 41.

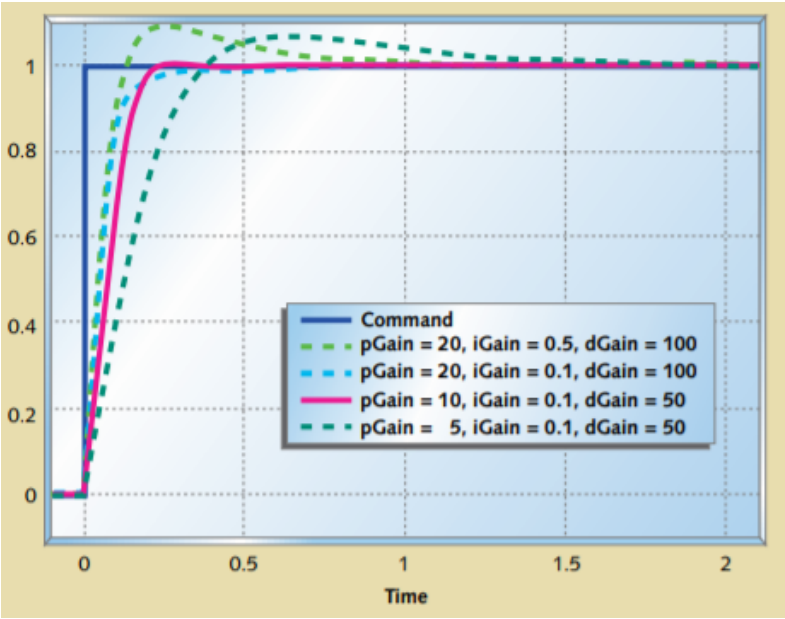


Figure 41: Impact of using P, I, and D terms [29]

3.6 Sine Wave Amplitude Calculation

The Direct-Quadrature (DQ) Transform

In the feedback loop of Figure 38, the AC/DC conversion stage can be done in either hardware or software. This and the following sections will discuss two methods of calculating amplitude that are typically (but not necessarily) done in software. For most electronics projects, this is usually done in a microcontroller.

The Direct-Quadrature (DQ) transform, proposed by several research papers, is adapted from the more commonly used three-phase ABC-DQ0 transform[30]. The fundamental property of the single phase transform is to transpose a sinusoidal signal onto a rotating coordinate system, in which a pure sine wave of constant frequency and amplitude will have a constant value. The axis is rotated so that θ, the current phase angle of the sine wave, is also the angle between the "x" axis (also known as the α axis) and the q axis [30]. This concept is shown in Figure 42.

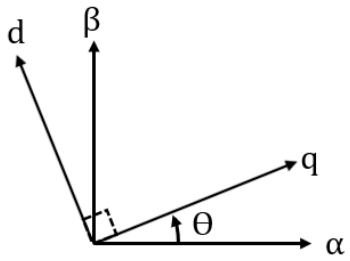


Figure 42: DQ transform rotating coordinate axis (labeled q and d) [30]

In the x-y or α-β plane, a cosine wave is represented by its real (horizontal) and imaginary (vertical) components. When its phase is 0 or π, the cosine wave is exclusively real; when its phase is $\pi/2$ or $3\pi/2$, the cosine is exclusively imaginary (and thus its real component is 0). In the dq plane, the cosine is at all times exclusively real.

In practical terms, the amplitude of a sine wave can be calculated using the DQ transform. However, the transform equation requires two waveforms. To obtain an entirely real answer (q $\neq \theta$, d $= \theta$), the waveforms must be a pure sine and cosine of the same phase angle (i.e two sine waves with the same amplitude and a 90 degree phase shift). In a single phase system, only one sine wave is available, so an orthogonal sine wave must be generated for processing [30]. This is shown in Figure 43 below:

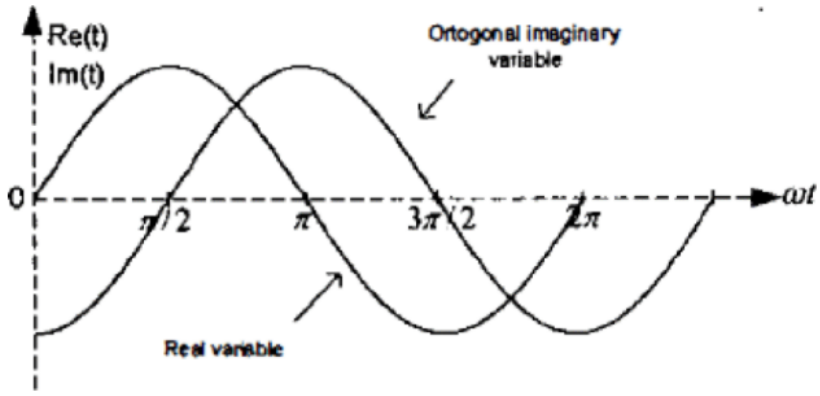


Figure 43: Real and "imaginary" sine waves, with 90 degree phase shift [30]

The "imaginary" sine wave can be created by using previous samples of the real variable so that they have a 90 degree shift. Mathematically, the amplitude can be calculated using the following formulae:

$$V_{\mathrm{dq}} = T v_{\mathrm{n}} \tag{4}$$

where V_{dq} is a 2x1 vector representing the d and q components of the output:

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix}$$

and where v_n is a 2x1 vector representing the instantaneous value of the sine and imaginary sine components:

$$\begin{bmatrix} v_{actual} \\ v_{imaginary} \end{bmatrix}$$

Finally, T is the transform vector, which is a function of the current phase of the real sine wave:

$$\begin{bmatrix} sin(\Theta) & -cos(\Theta) \\ cos(\Theta) & sin(\Theta) \end{bmatrix}$$

I/Q Sampling

Another method for detecting the amplitude of a sine wave is known as I/Q sampling. I/Q sampling represents the real and imaginary components of a cosine as a function

of time. It is most commonly used to find the magnitude and phase of each sinusoidal component of a modulated radio frequency (RF) signal. However, in its most basic form, it can be used to find the instantaneous amplitude of a single cosine wave, even if the wave varies in frequency or amplitude.

Like the DQ transform, I/Q sampling is based on the representation of a sinusoid in the complex plane. As shown in Figure 44, the real component of the sine wave is known as the I component, and the imaginary component is known as the Q component. At the instant in time represented by the figure, the sinusoid represented would have a phase angle of θ such that the real component was neither at its maximum nor at 0 [31].

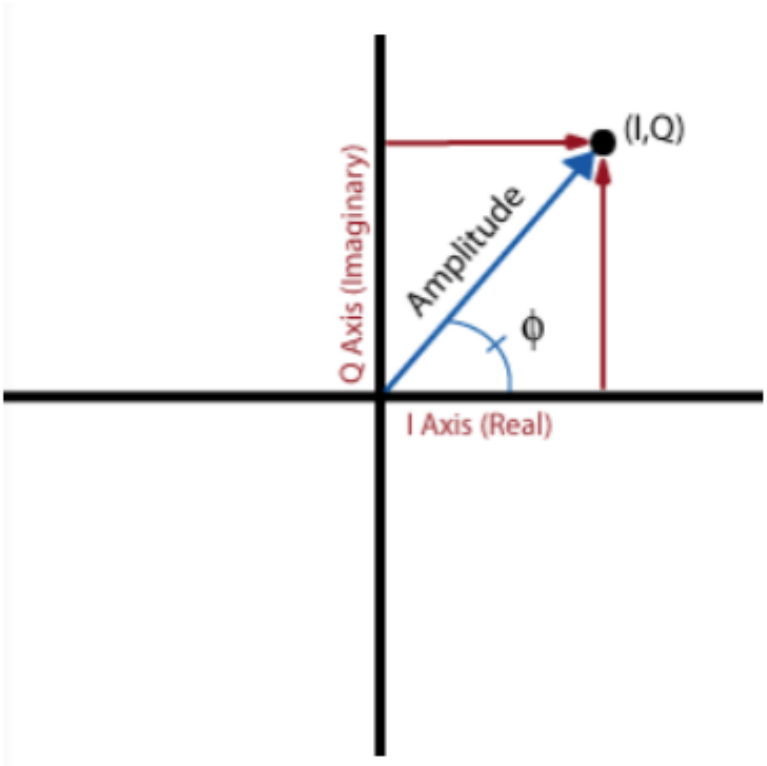


Figure 44: I and Q components of sinusoid [31]

Because the Q component is imaginary, its value is not directly evident from a voltage measurement of the signal. For a pure sinusoid (of constant frequency and amplitude) however, the Q component is the value of the signal phase shifted by -90°. A discrete sample of points for a pure sinusoid of amplitude 1 will draw a perfect circle in the IQ plane if samples taken at different points in time are plotted, as shown in Figure 45. The circle has a constant radius, which represents the constant amplitude of the sine wave [31].

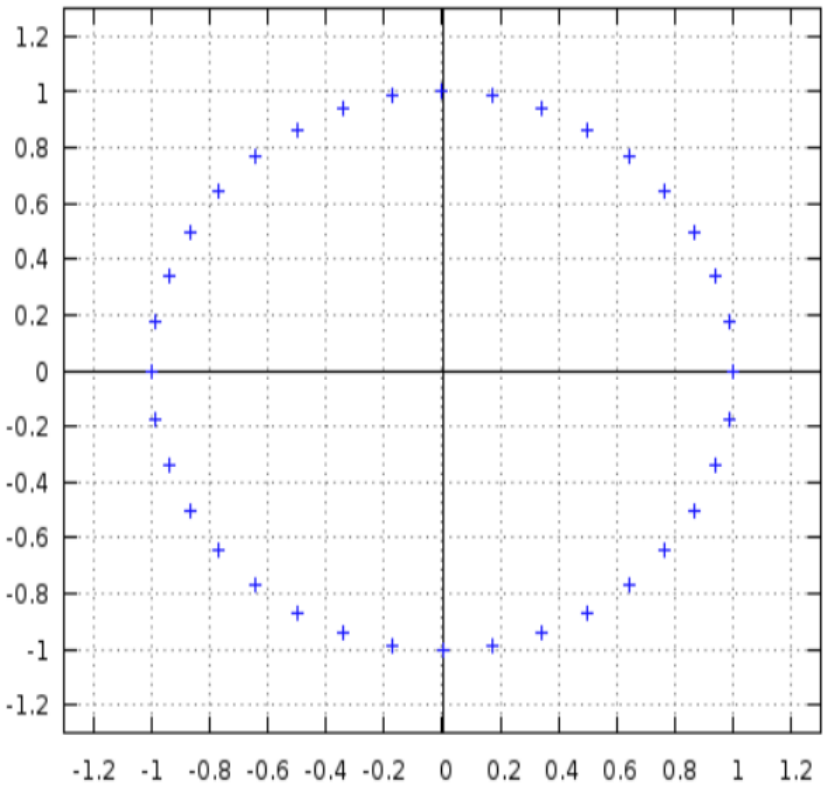


Figure 45: I/Q two-dimensional plot of sinusoid [31]

For our purposes, this is a significant enough depth of research into IQ sampling, because by using the I and Q components, one can immediately calculate the amplitude of a sine wave. Assume that the sinusoid in question has an amplitude of a, and that it is possible to create a second version of the signal that is phase shifted by -90°. We can then use the elementary algebra equation:

$$a^2 = (a * \sin(\theta))^2 + (a * \cos(\theta))^2 \tag{5}$$

We can easily solve for the amplitude (regardless of current phase angle), as shown below:

$$a = \sqrt{((a * \sin(\theta))^2 + (a * \cos(\theta))^2)} \tag{6}$$

At any point in the IQ plane, this is equivalent to saying:

$$a = I^2 + Q^2 \tag{7}$$

I/Q sampling has more advanced applications in determining the frequency and relative phases of sinusoidal components of a modulated radio frequency signal that are outside the scope of this report. Regardless, it is a simple method for determining the amplitude of a pure sine wave.

3.7 Safety Standards for Off-Grid Inverters

Like all consumer devices, off-grid inverters must be certified for safety before being sold. Two important standards bodies are Underwriter's Laboratories (UL) and the Institute of Electrical and Electronics Engineers (IEEE). UL publishes and maintains

Standards for Safety and standardized tests used to verify the safety of different types of products. If a product carries the UL Mark, Underwriters Laboratories found that samples of the product met UL's safety requirements.

There are two UL standards that are related to off-grid inverters. The first one is UL 1741, which is a standard that covers inverters, converters, charge controllers, and interconnection system equipment (ISE) intended for use in stand-alone (not grid-connected) or utility-interactive (grid-connected) power systems. One relevant part of this standard is that it requires galvanic isolation between input and output of the device. It also gives standards for the output power characteristics of the device, such as the allowable harmonic distortion [32].

Another standard is UL 458, which is a standard for Power Converters/Inverters and Power Converter/Inverter Systems for Land Vehicles and Marine Crafts. In addition, UL 458 also covers fixed, stationary and portable power inverters and power-inverter systems having a DC input and a 120 or 240 V AC output. UL 458 gives standards for grounding, transformers, fuses and other protective devices, etc [33].

IEEE-SA (The Institute of Electrical and Electronics Engineers Standards Association) is an organization within IEEE that develops global standards for eletronics and technology. Some related IEEE standards are IEEE C62.41.2, which is a guide for Array and Battery Sizing in Stand-Alone Photovoltaic (PV) Systems and IEEE C37.90.2 – EMI, which is the Standard for Withstand Capability of Relay Systems to Radiated Electromagnetic Interference from Transceivers.

4 Methodology

This section covers the major design considerations we made in developing our inverter. In this section, we discuss why we chose the low frequency architecture with a 3-level PWM H-Bridge topology for our inverter. We will also discuss how and why we chose our major components, such as our H-Bridge MOSFETs, MOSFET drivers, transformer, filter, etc. This section also covers the design of our voltage amplitude regulation feedback loop and our planned circuit testing methodology.

4.1 Inverter Architecture

After exploring different transformer architectures, we decided to use the Low Frequency Transformer architecture. When considering which transformer architecture to choose, we rated time and design risk to be our top priorities. Our goal was to have a complete, functioning prototype well before the end of C Term. As explained in the Background section, the high frequency transformer and transformerless architectures presented a significantly greater design risk than the low frequency topology. Additionally, we were advised by several knowledgeable people to use the low frequency architecture, because designing the inverter circuit would be difficult enough without the extra load of designing a DC/DC converter.

When making this decision, we understood that the low frequency architecture was ranked the lowest in efficiency, size and cost. Low frequency inverters tend to be relatively large and costly and rated at 80-90% efficiency. However, we determined that ensuring we would be able to build a functioning inverter was more crucial than maximizing efficiency and minimizing cost and size.

4.2 DC/AC Topology Selection

For the DC/AC topology, we chose a 3-level PWM topology with an H-bridge switching circuit. Multilevel inverters contain more components and much more involved circuitry than 2-level and 3-level inverters. They also require more complex control because they have more switches. Simpler multilevel inverters with few stages create lower frequency harmonics that are more difficult to filter; multilevel inverters with more stages require more components and complex control. Multilevel inverters are useful in high voltage applications due to the fact that they are able to split voltage between multiple switches; however, our inverter is operating at no more than a 15V input, for which transistors can easily be purchased. Thus, the additional complexity of a multilevel inverter was not justified for our project.

With a PWM topology, either 2 or 3-level PWM is possible, and both use the same H-Bridge circuit. 3-level PWM has a slightly more complex switching pattern than 2-level because it requires both high frequency switching and a low frequency square wave to drive the other two switches. However, this difference is relatively minor, and 2-level PWM theoretically has larger total harmonic distortion (THD). When 3-level PWM is used with square wave voltage control, switching losses can also be reduced, because 2 switches operate at 60Hz instead of at high frequency. For example, with 2-level PWM and 4 switches operating at 25kHz, each switch is turned on 25,000 times per second and turned off 25,000 times per second. Thus, the losses in the circuit are equal to:

$$4 * (2 * 25000) * (P_t) = 200000 * P_t$$

where P_t is the power dissipated in a single switch transition. For a 3-level PWM system with 2 switches at 60Hz, the losses are effectively only half as much:

$$(2 * (2 * 25000) + 2 * (2 * 60)) * (P_t) = 100240 * P_t$$

4.3 MOSFET Selection

We decided to use four N-Channel MOSFETs for our H-Bridge. This decision was motivated by the simplicity of choosing and testing only one type of MOSFET, instead of choosing both an N-Channel and P-Channel MOSFET. To control a half bridge with a high-side N-Channel MOSFET, however, requires a bootstrapping circuit to drive the high-side gate, as the high-side source is not grounded. While a bootstrap circuit could add additional complexity to the H-Bridge, many MOSFET drivers are available with integrated bootstrap circuits, eliminating any additional design complexity.

We chose to use a MOSFET instead of a GaNFET for several reasons. First, using a device with which we are familiar (MOSFET) and which is explored more thoroughly in literature lowers our design risk. Second, because we chose to switch in the kHz range, the fast switching times of the GaNFET were not required. Third, we did not believe the potential of higher efficiency by using GaNFET would justify the significantly higher cost.

Our MOSFET selection was based primarily on choosing a MOSFET with an acceptable maximum drain source (breakdown) voltage and maximum drain current while minimizing the drain-source on-resistance. We decided to use a design margin of 200% - i.e. when possible, components should be rated for at least twice their operating current and voltage. Thus, the breakdown voltage must be rated at 200% of the expected drain-source voltage, and the maximum drain current should be twice the current we plan to switch. We only evaluated MOSFETs that met these criteria. With a maximum input voltage of 15V, we rated our MOSFETs to have a breakdown voltage of at least 30V. Our current rating for the load was about 10A so we chose MOSFETs that had current ratings of over 20A.

The main criteria for choosing the best possible MOSFET for our application was minimizing losses. The most critical factor in minimizing losses is minimizing $R_{DS\text{-}ON}$, which has a linear relationship with conducting losses. Thus, this is given the largest weight (5) in our MOSFET value analysis in Table 46. Minimizing the turn-on and turn-off times is also important to reducing switching losses and maximizing the frequency at which the transistor can be switched.

MOSFET
Value Analysis

	Quality	Market Weight	PSMN017-30PL Value point	PSMN017-30PL Feature	PSMN017-30PL Total	IRL2703PbF Value point	IRL2703PbF Feature	IRL2703PbF Total	NTD4815N Value point	NTD4815N Feature	NTD4815N Total
1	RDS-ON	5	4	17mOhm	20	3	40mOhm	15	4	15mOhm	20
2	Turn on time	4	3	10.7ns	12	4	8.5ns	16	3	11ns	12
3	Turn off time	4	3	11.4ns	12	3	12ns	12	2	25ns	8
4	R_junction-ambient	2	4	60C/W	8	4	62C/W	8	3	100C/W	6
5	Total gate charge	3	3	10.7nC	9	2	15nC	6	2	14.1nC	6
6	Cost	2	3	$0.92	6	3	$0.95	6	4	$0.61	8
	Total										
	Totals:				67			63			60

Figure 46: MOSFET value analysis

Our highest score was the PSMN017-30PL MOSFET with a total of 67. However, although we had originally only considered MOSFETs with simulation models available online, we were unable to successfully implement the model for the PSMN017-30PL in Multisim. Thus, we chose the next highest scoring MOSFET, the IRL2703. We were successfully able to simulate with this MOSFET, as is discussed in the Simulation section of the report.

4.4 MOSFET Driver Selection

Justification for use of Driver

The purpose of a MOSFET driver in our circuit is to supply adequate voltage and current to the gate of each MOSFET. A microcontroller cannot, on its own, drive our H-Bridge for 3 reasons:

1. RDS for the IRL2703 (when operated as a switch) is specified as low as 40mΩ, but only if VGS reaches 10V or more. With microcontroller output voltage limited to 3.3V maximum, the maximum conducting efficiency of this MOSFET cannot be realized.

2. The high side NMOS on each side of our H-Bridge is floating, and has its source referenced to the drain of the low side NMOS (instead of being referenced to ground). Thus, simply applying a constant voltage to the gate of the high side switch is not sufficient to turn the device on and maintain operation in the triode region. This will be explained in more detail in the following section

3. The current required to turn on the IRL2703 can be estimated based on the desired (maximum) turn on time and the total gate charge, based on the following equation:

$$I = Q/t$$

   Solving for I, we use Q = 15nC (total gate charge of IRL2703) and t = 8.5ns (turn on delay time), and find that a fair estimate of the average turn on current for the IRL2703 is 1.75A. Our MOSFET driver should be able to source/sink this current or more, in order to minimize switching losses. Most microcontrollers can output only 10's or 100's of mA from each pin, which is far too low to turn the MOSFET on at high frequencies.

Driver Selection Criteria

Half-bridge drivers are widely available for applications similar to ours, and thus we will consider a driver that controls a half-bridge (both MOSFETs on one side of the bridge). We refer to these as the high side (SW1) and low side (SW3) switches, where SW2 and SW4 make up the other half bridge. As shown below, the voltage between them is labeled as V_{S1}, or the source voltage of the high side switch. The figures in this section were built by the team in simulation software and are shown in Figure 47 (a)-(d).

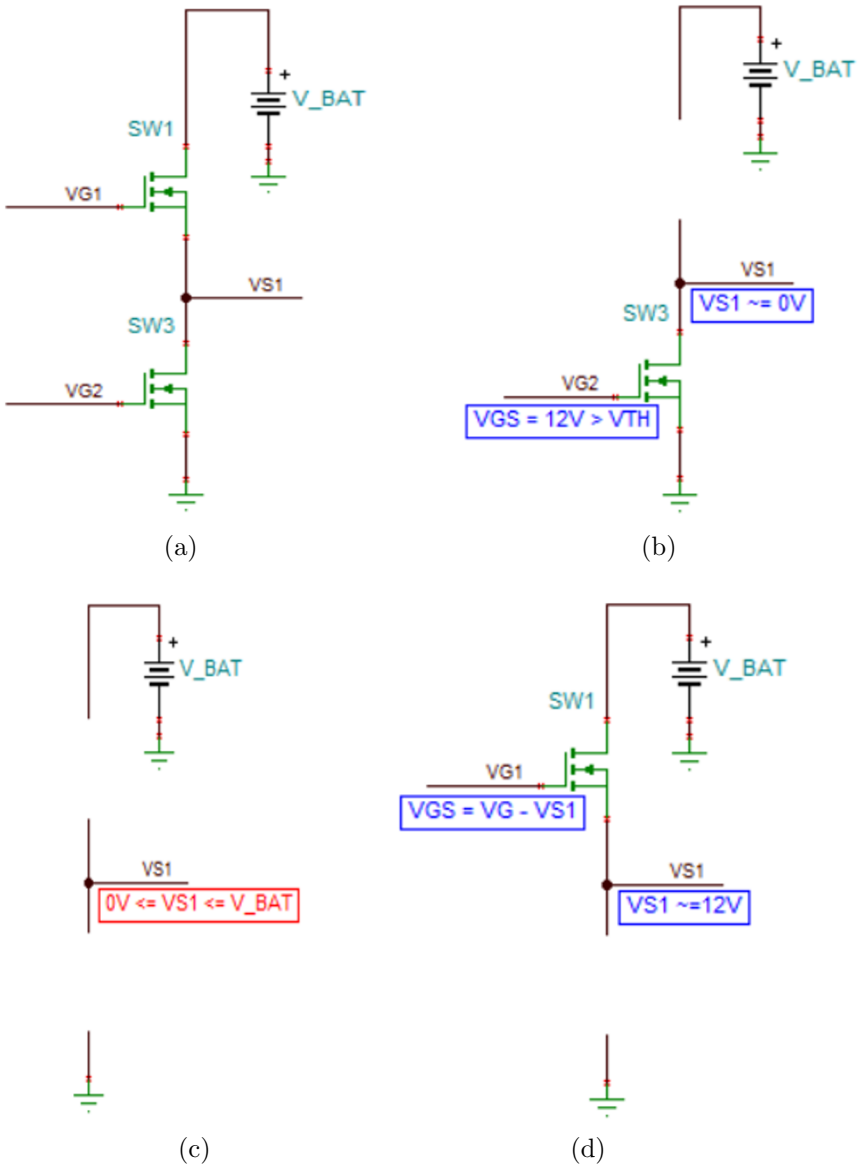


Figure 47: Half-Bridge switch positions: (a) Half-Bridge, (b) High-side switch off, (c) Both switches off, and (d) Low-side switch off

When the low side MOSFET is turned on, the circuit appears as in Figure 47(b), where the high side switch is turned off and is essentially an open circuit. The low side gate voltage must simply exceed 0V + V_{TH} in order to turn on the switch ($V_{GS} > V_{TH}$).

Before turning on the high side switch, the driver must then turn off the low side switch, as shown in Figure 47(c). At this point, the source voltage for the high side is floating somewhere between 0V and V_{BAT}. Ideally, it would be equal to $0.5V_{BAT}$, but in a situation with fast switching, the voltage may not stabilize at this level before the high side switch is turned on. Also, it is unlikely that the two MOSFETS will have exactly equal resistance when they are turned off.

Thus, it is clear that to turn the high side switch on (Figure 47(d)), its gate voltage must be equal to at least $V_{S1} + V_{TH}$. As the high side switch begins to turn on, VS1 quickly rises to be approximately equal to V_{BAT}. Thus, the driver must be able to apply $V_{TH} + V_{BAT}$ to the high side gate.

V_{BAT} is typically the largest DC voltage available in a circuit. Thus, special circuitry

is needed to add the desired V_{GS} to V_{BAT}. Half bridge drivers are available that use a "bootstrapping" circuit specifically for this purpose, so our driver search was limited to these drivers. Other important criteria were:

Category	Value	Justification
Sink/source current	> 1.75A	Previously mentioned, 1.75A is expected average MOSFET gate current during turn on
Supply Voltage	10-15V	12V battery is the supply used for driver. Driver must be able to operate on 10-15V and supply up to 15V to gate of low side MOSFET
Bootstrap Voltage	≥ 30V	When battery voltage is at a maximum (15V), while turning on high side, V_{S1} max = 15V, so gate voltage will exceed this voltage by V_{BAT} (15V). Thus, $V_S = 15$, $V_G = V_S + 15V = 30V$
Logic Levels	3.3V logic compatible	Microcontroller will operate with 3.3V supply
Rise/Fall time	As fast as possible	Take full advantage of short rise/fall times of MOSFET
Simulation Model	Must be available	Must simulate behavior to ensure compatibility with MOSFET

Table 13: Additional requirements

Based on these criteria, we determined that the Texas Instruments UCC27201 was a suitable driver for our application. The UCC27201 is a 3.3V logic-level driver that can source/sink up to 3A, operates with a supply voltage of 8-17V, and can bootstrap up to 110V. It has a rise time of 8ns and a fall time of 7ns. Most importantly, it was one of the few suitable drivers we found that had a readily available PSpice model.

4.5 Transformer Selection

The transformer for our chosen architecture is used to step battery voltage up to mains voltage AC. In our case, the minimum AC input to the transformer will be $10V_{Peak}$ ($7V_{RMS}$), and the desired output is $162V_{Peak}$ ($115V_{RMS}$). The full criteria are shown in Table 14.

Category	Value	Justification
Input Voltage	$7V_{RMS}$	Minimum voltage to be input to transformer is $7V_{RMS}$. Our feedback system can scale higher voltages down to 7V, but we cannot easily increase the minimum voltage, so 7V is our nominal input
Output Voltage	110-120V_{RMS}	Standard range of specifications for line voltage
Current Capability-Primary	$17.2A_{RMS}$	For nominal power of 60W and nominal voltage of $7V_{RMS}$, P = I*V gives $8.6A_{RMS}$. With a design margin of 2, must be able to handle 17.2A
Current Capability-Secondary	$1A_{RMS}$	For nominal power of 60W and nominal voltage of $120V_{RMS}$, P = I*V gives $0.5A_{RMS}$. With design margin of 2, must be able to handle 1A.
Power capability	120VA	With a nominal load of 60W and a design margin of 2, the transformer should be rated at 120VA or more
Cost	Minimal	60Hz transformers are notoriously large compared to high frequency transformers, and thus more expensive
Size	Minimal	As with cost, 60Hz transformers can be large and heavy - minimizing this makes our final product more viable

Table 14: Transformer selection criteria

After searching for 60Hz transformers with these characteristics, we could not find a readily available step up transformer matching our specifications. Thus, we were forced to consider using a step down transformer in reverse. We chose the Triad Magnetics F-22A, which is a chassis mount 60Hz transformer. It has a nominal primary voltage of $115V_{RMS}$, nominal secondary voltage of $6.3V_{RMS}$ at $20A_{RMS}$, and can handle a maximum power of 126VA. It weighs 7lbs and costs upwards of $35. We discussed our project with a Triad Magnetics application engineer, who was able to test the transformer for us and ensure that it worked properly in reverse.

4.6 Microcontroller Selection

Our microcontroller is meant to serve two functions: first and most importantly, reliably generate PWM; and second, take in a feedback signal and use it to regulate output voltage. Our selection was influenced by the 2012 MQP, "Three-Level PWM DC/AC Inverter Using a Microcontroller" [5], which recommended using a microcontroller with the ability to generate faster PWM than the Texas Instruments (TI) MSP430. We searched for a faster microcontroller that was also specifically marketed towards power electronics and controls, and settled on the TI C2000, which is a 32 bit microcontroller that is marketed specifically towards control and power electronic applications. We specifically chose to buy the LaunchXL-F28027, a LaunchPad development kit with the TMS320F28027 microcontroller, which is the most basic C2000 launchpad available.

The TMS320F28027 operates at 60MHz, which is a significant speed upgrade from the 16MHz MSP430. It has 64kB of flash memory, which we believed to be adequate to hold our relatively small program. It has 8 PWM channels (twice as many as the 4 that

we need), which TI advertises as "Enhanced PWM" or ePWM. The ePWM channels can be configured to run entirely on hardware, with little or no software oversight. These pins are connected to digital outputs on the F28027 Launchpad. The C2000 also has a free library known as "IQ Math" which includes highly optimized fixed-point calculations that emulate floating point calculations with higher speed (such as trigonometry, exponentials, and floating point arithmetic). IQ Math is useful for efficiently conducting mathematically intensive work inside Interrupt Service Routines (ISRs) and other microcontroller tasks that must meet strict timing deadlines.

4.7 Output Filter Design

The purpose of the output filter is to convert 3-level PWM into a sine wave output after H-Bridge. The three main types of filter are the low pass filter, high pass filter and band-pass filter. A low pass filter is most useful for an inverter because high frequency PWM should be filtered out, leaving only the low frequency fundamental.

The low pass passive filter can be further subdivided into two different types: LC low pass filter and RC low pass filter. Because ideal capacitors and inductors do not dissipate any power, whereas resistors do, the LC low pass filter was the most suitable for our inverter.

The LC low pass filter can be designed as a first order, second order, etc. As the filter order increases, more capacitors and inductors are added to the filter network. Higher order filters are often desirable because they have a greater attenuation slope above their cutoff frequency. We decided that our design did not require a LC low pass filter design greater than the first order. As a result, our final filter design was a first order LC low pass passive filter. The filter design is discussed in more detail in the Simulation section, because it was iteratively designed through simulation.

4.8 Feedback Loop Design

Based on the (relatively) simple voltage control feedback loop discussed previously (shown again in Figure 48), we have decided to use a PID controller based feedback system to maintain a stable output voltage. This feedback loop is intended to maintain our output voltage well within the range of $115V_{AC}\pm10\%$ specified by our product specifications.

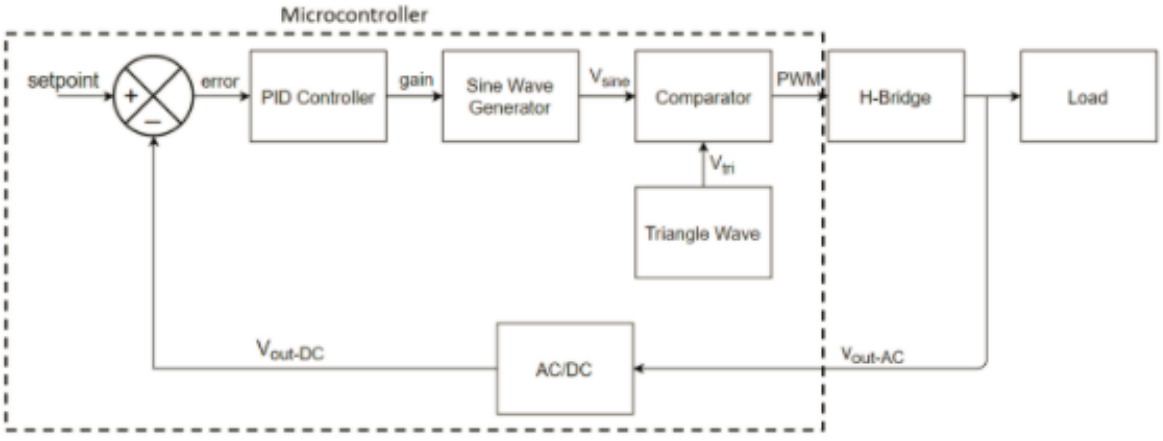


Figure 48: Feedback Loop Block Diagram

As our design specifies that we will use a microcontroller to generate PWM, we intend to also use the microcontroller as a primary component of the feedback loop. As shown in Figure 48, the AC/DC conversion stage, comparator, PID controller, and PWM generator will be integrated into the microcontroller. However, before passing the output voltage to our microcontroller, we will use a circuit to scale the output voltage to be within the acceptable range of the ADC unit on the C2000 (0 to 3.3V). The schematic of this circuit, which was adapted from the TI C2000 Single Phase Inverter kit schematic, is shown in Figure 49. This circuit is a differential amplifier with a gain of 11K/3M = 0.003666 and an offset of 1.65V. This will give the (nominal) $170V_{peak}$ output from the inverter a maximum ADC input value of 170 * 0.003666 + 1.65 = 2.27333V and a minimum ADC input value of -170 * 0.003666 + 1.65 = 1.02666V. These maximum and minimum voltages are well within the 0-3.3V range of the C2000 ADC.

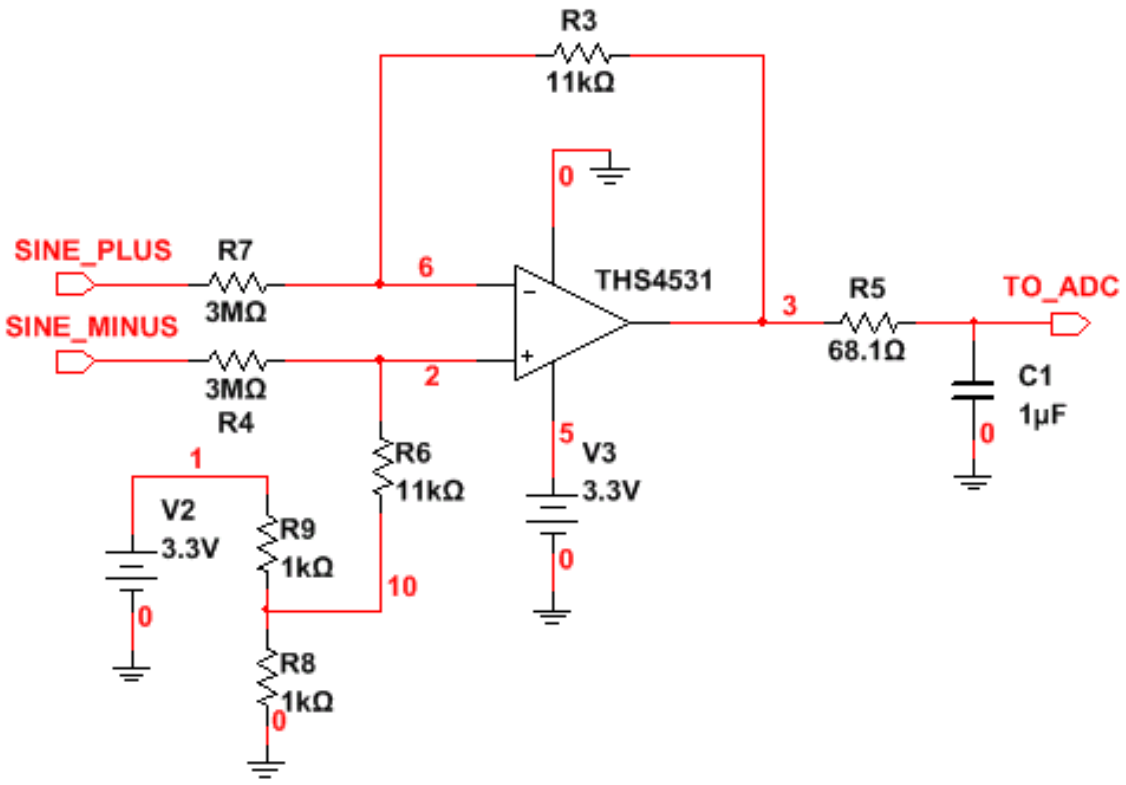


Figure 49: Feedback sampling circuit

The AC/DC conversion stage will be different than that tested in simulation (shown in Appendix B). The peak follower was shown to cause the load voltage to be rather unstable, and a discrete time solution that samples the amplitude of the sine wave more than once per cycle (as with a peak follower) can be implemented. The solution of choice is based off of I/Q sampling, which will provide a much more stable measure of amplitude.

The setpoint will simply be a software constant specifying the ADC reading that corresponds to the appropriate output voltage. The PID controller will be implemented in software as well.

The triangle wave generator will actually come from an up-down hardware timer in the ePWM module on the C2000. The sine wave generator will come from a sine wave

value calculated in real time and adjusted for every period of the timer. The ePWM module will be configured so that when the timer count crosses the "compare" count set by the sine wave value, the PWM output will toggle. A single PWM module will control the two complementary outputs used to switch the high frequency side of the H-Bridge. This is shown in Figure 50. In this figure, ZI points to the minimum value of the timer, CA (500) indicates the "compare" value used to trigger a change in output, PA (800) indicates the maximum value of the timer, RED is the rising edge delay (deadtime) and FED is the falling edge delay. All of these user configurable values allow for a variable frequency PWM signal with configurable deadtime and duty cycle.

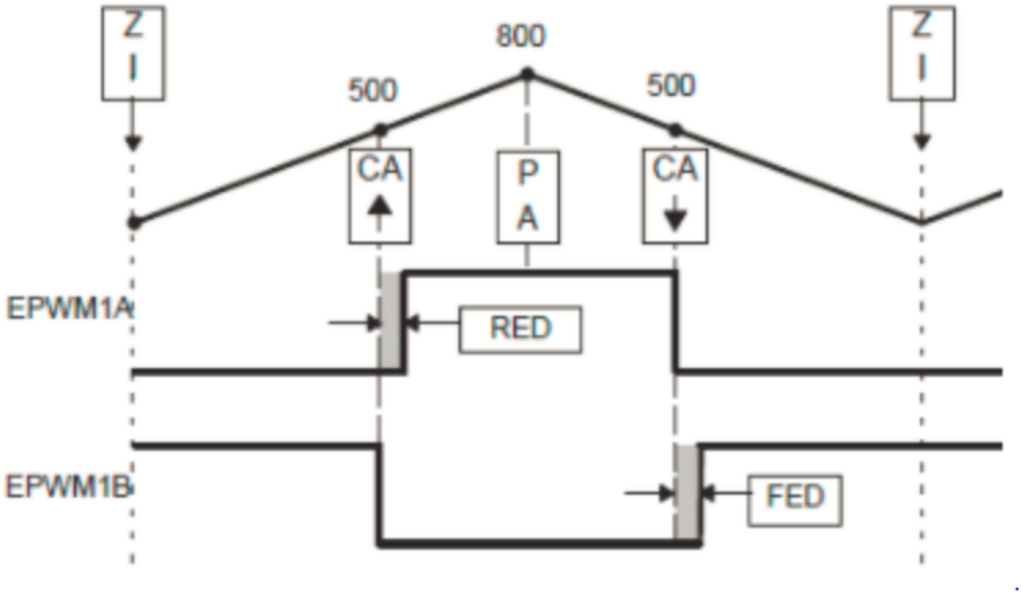


Figure 50: ePWM compare module operation [34]

4.9 Initial Schematic

Based on our major component selections, we developed the initial schematic shown in Figure 51, which would be updated as we refined our circuit through testing.

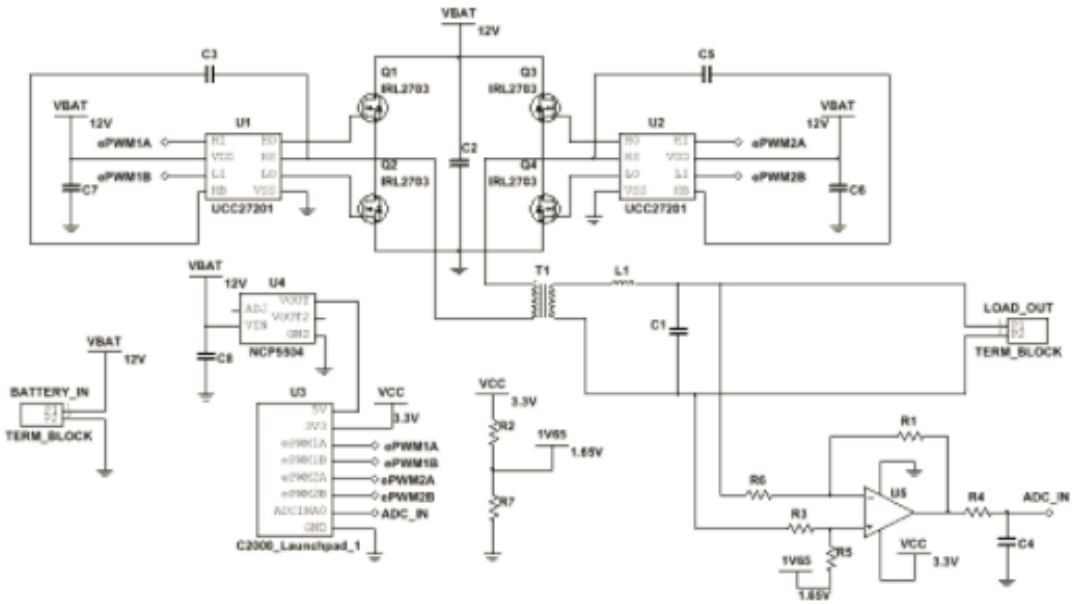


Figure 51: Initial circuit schematic, prior to simulation and testing

4.10 Circuit Testing Methodology

Our test plan was to, when possible, test each component on a breadboard before implementing the circuit on a PCB. We planned to test our transformer, MOSFET, driver, feedback circuit and low pass filter separately first to make sure that each part would work properly on its own. However, because the breadboard cannot handle high current and high frequency signals, a printed circuit board (PCB) is required for testing and implementing the full circuit. Thus, our second test stage is to test the whole inverter circuit on a custom PCB.

4.11 Microcontroller Power Source

Although our initial plan was to use a 12 to 5V linear regulator to power the microcontroller, we later realized that this was a short-sighted decision after determining that the current requirement of the C2000 Launchpad may be as high as 120mA or more [35]. Assuming a maximum input voltage of 15V, the linear regulator would have a voltage drop of 10V and a current (both input and output) of 120mA or more. In a worst case scenario, with I = 200mA, the power dissipation of the regulator would be approximately 2W. Assuming a 60W load, wiith 2W lost to power the MCU, efficiency would decrease by at least 3%, because 2W / (60W + 2W) * 100% = 3.23%. Thus, we decided to use a switched mode power supply (SMPS) to power the MCU.

Our chosen SMPS was the TI TPS5431, which we purchased as part of the prefabricated TPS5431EVM evaluation module. The TPS5431EVM has a rated input of 9 to 21V (although the TPS5431 chip itself can accept as little as 6V) and it uses a regulated buck topology to generate static output of 5V. It can output as much as 3A. Its efficiency varies with its output current - with a 15V input (worst case) and a very low current output (less than 200mA), efficiency may be as low as 85%. Using this worst case number, we can calculate the power dissipation in the SMPS by (1 - 0.85) * (15V

* 200mA) = 0.45W. This is far superior to the 2W that could have been wasted by the linear regulator, accounting for only 0.75% of our rated output.

4.12 Summary

This section discussed our key component selections and project methodology. Our inverter will use the low-frequency architecture with a 3-level PWM H-Bridge topology. Major components such as the 60Hz transformer, MOSFET, and driver were chosen according to this topology selection. Additional design decisions such as our choice of a microcontroller and feedback system were also discussed.

5 Simulation and Analysis

Our team chose National Instruments™ Multisim software to simulate out inverter circuit. Multisim is a Spice-based circuit simulator that is powerful enough to handle the relatively low frequency nature of our simulations, is able to import PSpice component models (which are often provided by manufacturers), and is taught at WPI, so knowledgeable people would be available to help us. In addition, another reason that our team chose Multisim is that WPI students have free access for Multisim.

5.1 H-Bridge and Filter Simulations

The circuit in Figure 52 is our original simulation circuit used to test the H-Bridge circuit and filter. All of the components are Multisim models. On the left side of the H-Bridge, a comparator is used to compare the magnitude of the switching frequency triangle wave and a 60Hz frequency sine wave. When the triangle wave is larger than the sine wave, the output of the comparator will go high to 12V, otherwise the output will go low to 0V. This is the drive signal for high-side switch Q1. However, it must be "bootstrapped" in order to correctly power the switch because the source of Q1 is floating. Thus, the gate cannot just be driven by a voltage $V_G > V_{TH}$, but instead by a voltage $V_G > V_{TH} + V_S$. Thus, the summing block A1 is used to add the gate drive voltage (V_{G1_drive}) to the source voltage (V_{PRI_LOW}). The non-bootstrapped gate drive voltage (V_{G1_drive}) is simply inverted (through a comparator) to drive Q2. For the other half bridge, a 60Hz square wave is used to directly drive the high side switch, Q3, which is also bootstrapped. This signal is inverted through a comparator to drive Q4.

When Q4 is driven low and Q3 is driven high, Q3 is a short circuit and Q4 is an open circuit, and V_{PRI_HIGH} will be 12V. During this time, Q1 and Q2 are alternately switched at high frequency. The voltage between V_{PRI_HIGH} and V_{PRI_LOW} can either be 0V or -12V. When Q3 is off and Q4 is on, Q4 will short the ground, and the point V_{PRI_LOW} will be 0V. During this time, Q1 and Q2 are still alternately switched at high frequency. The voltage between point V_{PRI_HIGH} and V_{PRI_LOW} can either be 0V or +12V. Therefore, the output voltage after the H-Bridge can be either +12, -12, or 0V. After being stepped up through a transformer (T2), an LC low-pass filter is used to turn the voltage into an approximate sine wave.

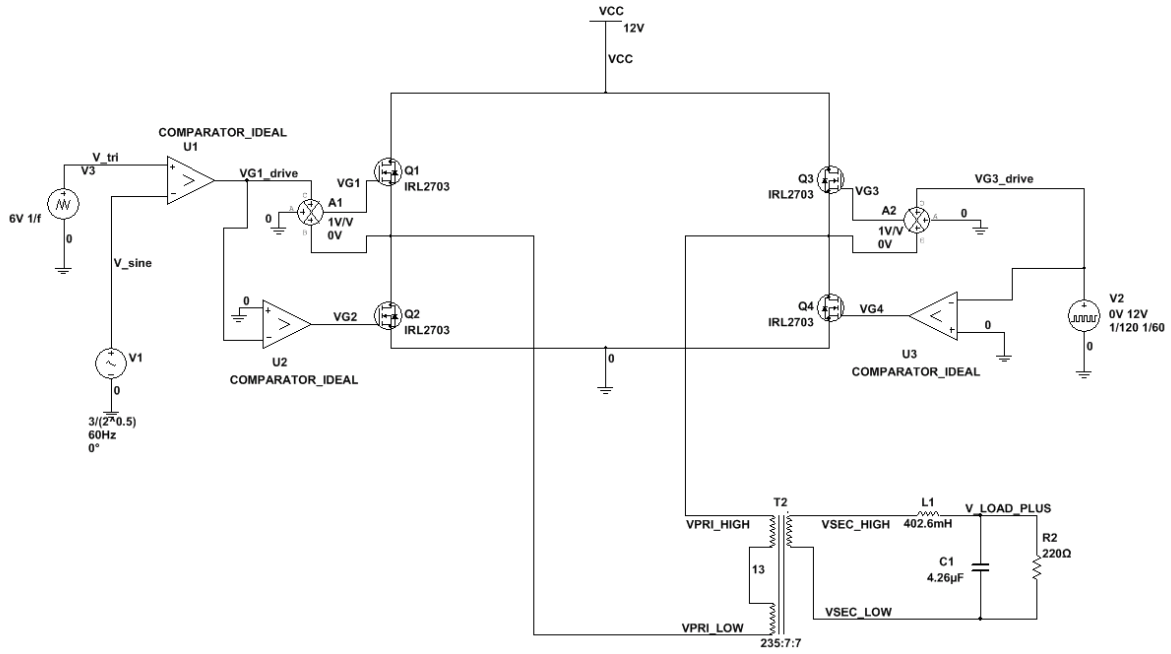


Figure 52: Simulation circuit with transformer and low pass filter

Figure 53 is the process of generating PWM for driving a transistor. The sine wave is 60Hz frequency, the triangle wave is 300Hz (5 times the fundamental frequency), and the square wave is the output of a comparator which has the sine and triangle waves as inputs. PWM generated from a switching frequency that is an odd multiple of the fundamental frequency is symmetric around the peaks of the sine wave.

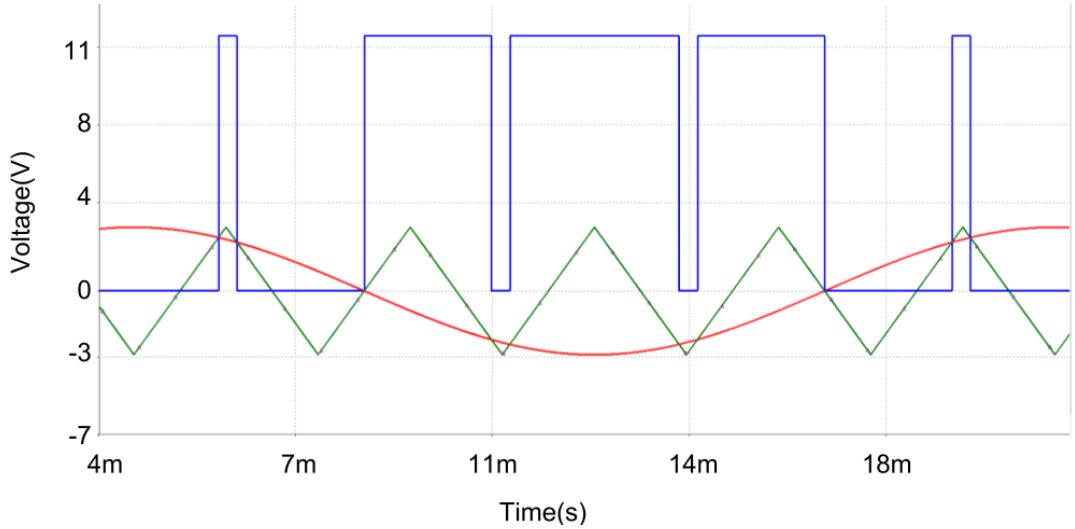


Figure 53: PWM generation with a frequency of 300Hz (5 times the fundamental frequency) (from simulation)

Figures 54 and 55 show 3-level PWM, which is generated from the H-Bridge. Figure 54 shows the 3-level PWM generated at 5 (odd) times the fundamental frequency, which is 300 Hz. Figure 55 shows the 3-level PWM generated at 6 (even) times the fundamental frequency, which is 360 Hz. As the figures suggest, 3-level PWM from 300 Hz (odd multiple of fundamental) switching frequency is symmetric, while 3-level PWM from 360

Hz switching frequency (even multiple of fundamental) is not symmetric around the peaks of the sine wave.

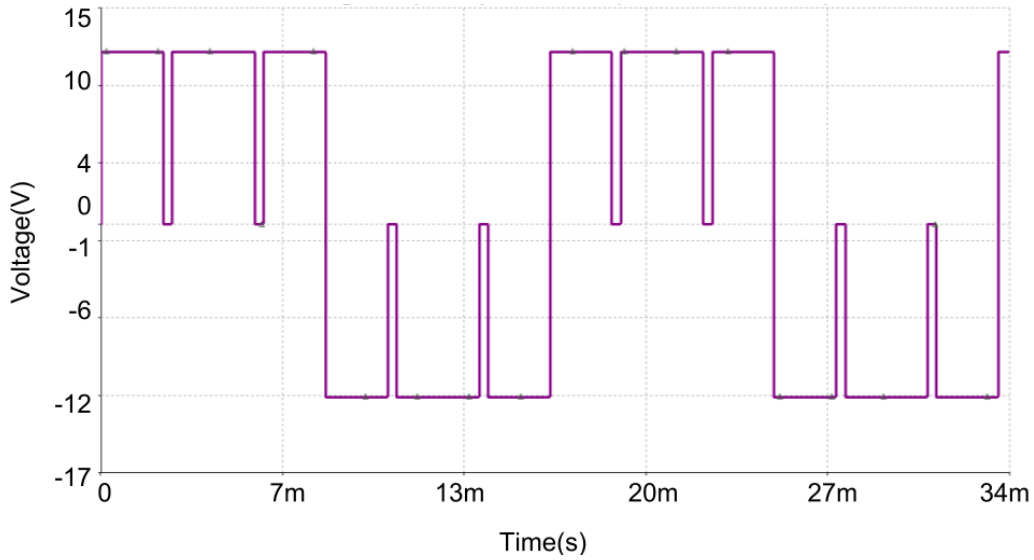


Figure 54: 3 level PWM at 300Hz (from simulation)

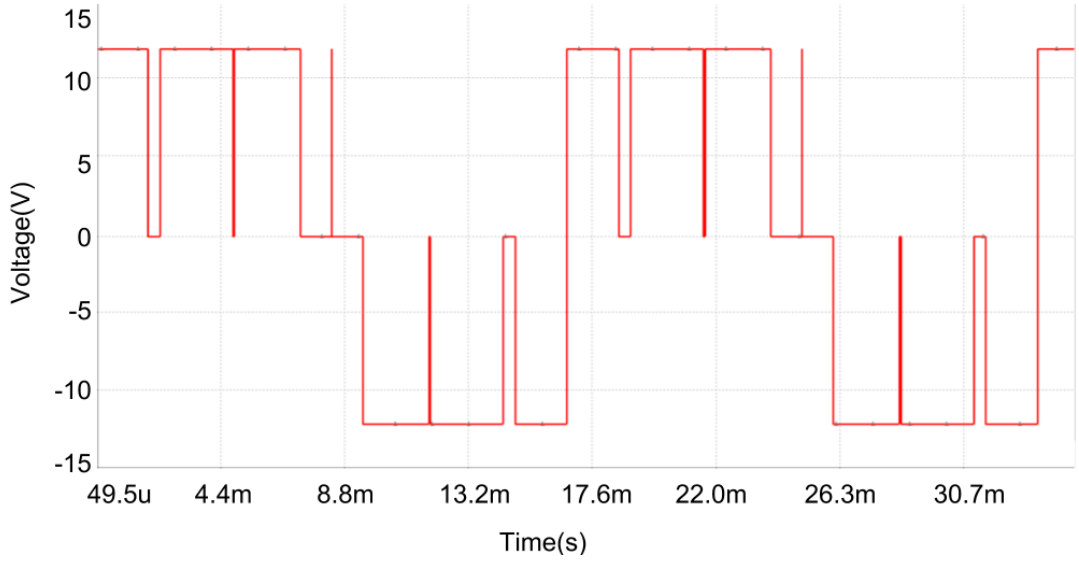


Figure 55: 3 level PWM at 360Hz (from simulation)

After generating 3-level PWM, the next step is to convert the PWM into a sine wave output. Figure 56 is the Bode plot of the LC low pass filter with 115Hz cut off frequency (f_{3dB}) which our team designed. A series inductor and a shunt capacitor are used in the low pass filter circuit. It can be seen here that when frequency = 60Hz, Vout = 0dB, which means Vout/Vin = 1. We calculated gain(dB) by the equation:

$$Gain(dB) = 20 * log(\frac{V_{out}}{V_{in}}) \tag{8}$$

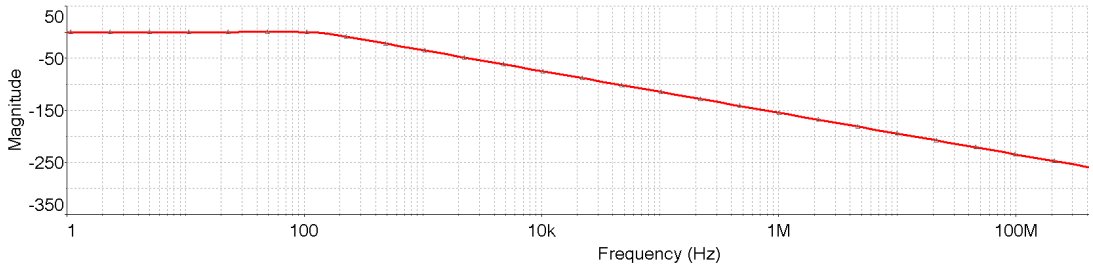


Figure 56: Simulated Bode Plot of our filter design

By using the low pass filter above, our team successfully generated a sine wave from 3-level PWM. Figure 57 is the sine wave generated from 300Hz switching frequency, which is 5 (odd) times the fundamental frequency. Figure 58 below is the sine wave generated from 360Hz switching frequency, which is 6 (even) times the fundamental frequency. Although 3-level PWM from 300 Hz (odd) switching frequency is symmetrical, unlike that from 360Hz (even) switching frequency, the interesting thing is that the sine wave generated from 360Hz (even) has lower THD.

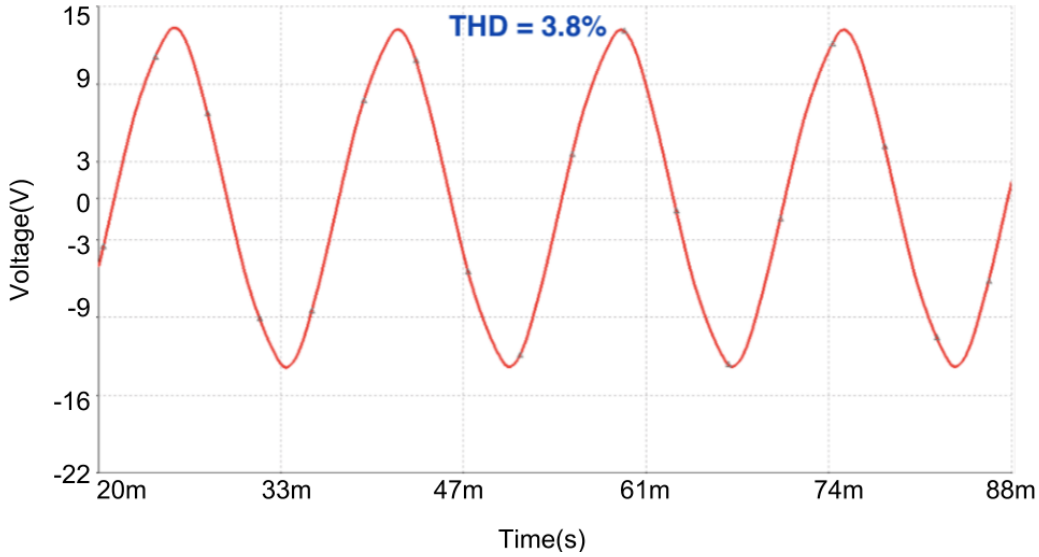


Figure 57: Filtered sine wave with 300Hz switching frequency (from simulation)

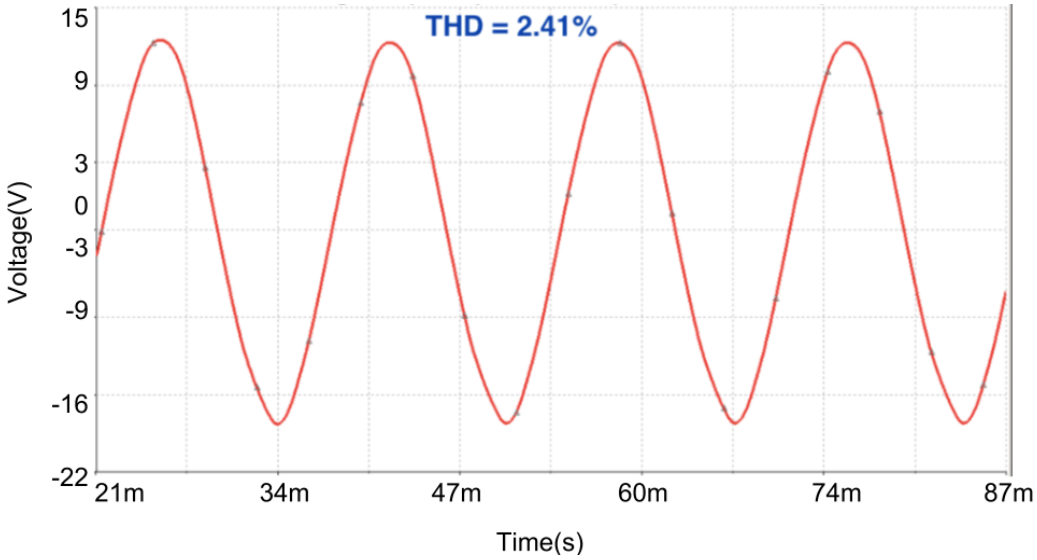


Figure 58: Filtered sine wave with 360Hz switching frequency (from simulation)

Inverters are typically switched in the kHz range, so these low frequency simulations simply show the concept of PWM. As we can see in Figure 59, the sine wave output becomes smoother at higher switching frequency (here it is 24kHz). This is because more pulses are generated with a higher switching frequency, increasing the accuracy of the sine wave. Our team discovered that when the switching frequency is in the kHz range (1kHz - 99kHz), the THD of the output sine wave does not change much.

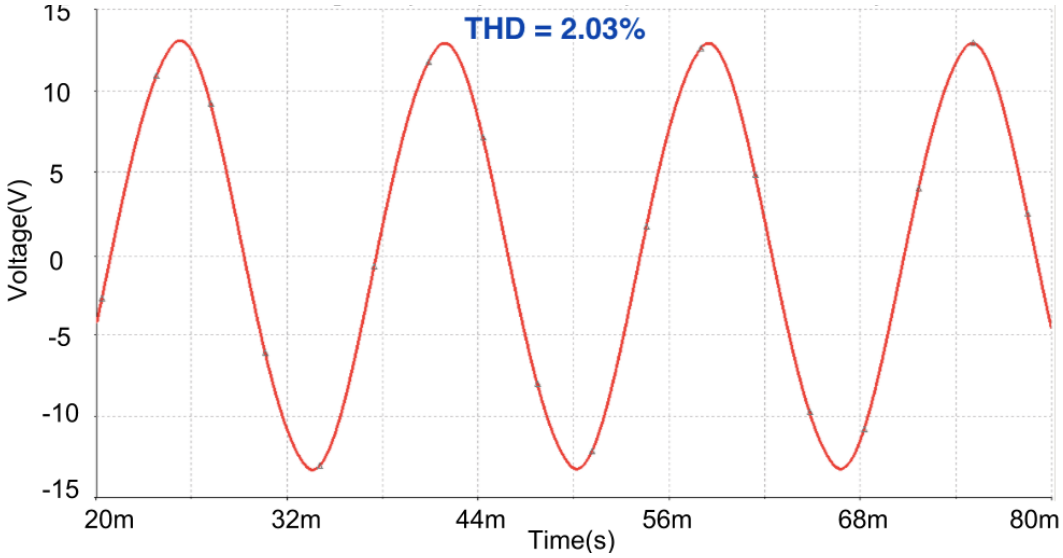


Figure 59: Filtered sine wave with 24kHz switching frequency (from simulation)

Using Multisim's Fourier Analysis feature, we took an FFT of the waveform. The % THD from each individual harmonic (2nd through 5th) is shown in Table 15.

	2nd Harmonic	3rd Harmonic	4th Harmonic	5th Harmonic
THD	0.23%	2.08%	0.85%	0.7%

Table 15: THD for 24kHz switching frequency (from simulation)

When switching frequency/fundamental frequency ratio is smaller than 21 times, we found that THD (after filtering) at even frequencies is only 2.41%, which is lower than that at odd frequencies (3.8%). However, when switching frequency/fundamental frequency ratio is bigger than 21 times, THD does not change much, regardless of whether the switching frequency is an odd or even multiple of the harmonic. For example, the THD at 24 KHz (400 times the fundamental) is 2.026% and THD at 24.06 KHz (401 times the fundamental) is 2.027%. 9 harmonics (Frequency range from 0Hz to 9*60Hz) were used to calculate the THD.

We also conducted some simulations with a 2-level PWM output. Figure 60 shows the PWM generated with a switching frequency of 300Hz and Figure 61 shows the filtered sine wave.

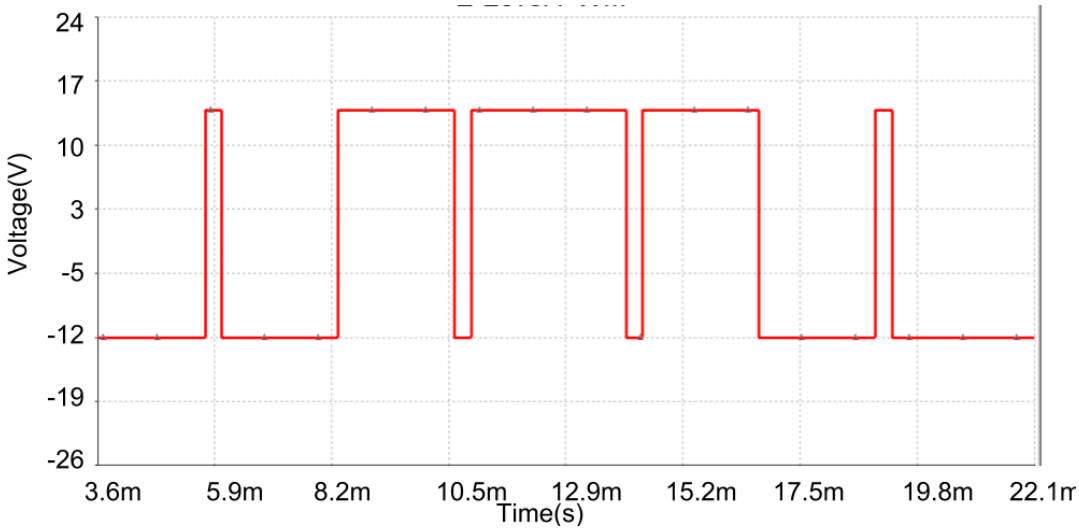


Figure 60: 2-level PWM at 300Hz (from simulation)

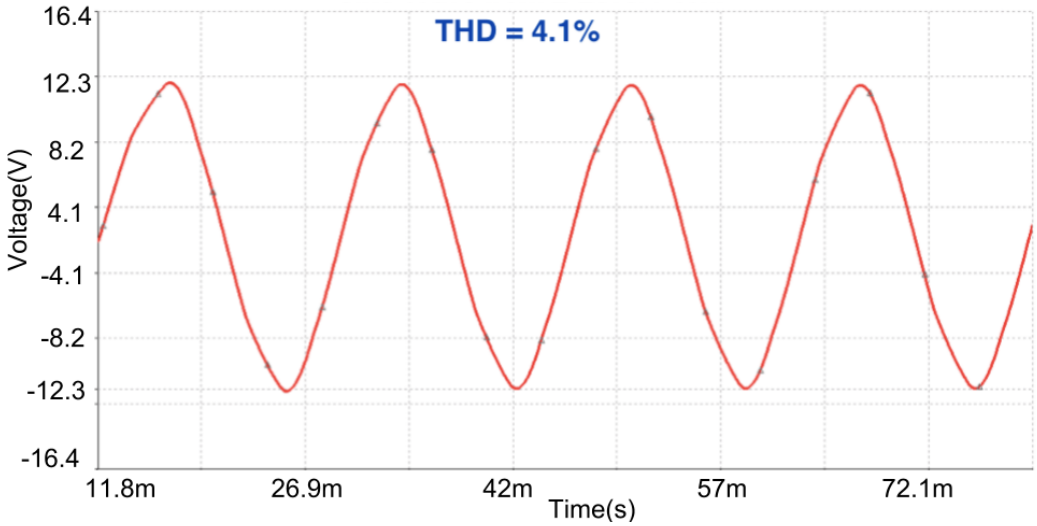


Figure 61: Filtered 2-level PWM at 300Hz (from simulation)

Fourier Transform simulations show that THD in 2-level PWM (4.1%) is slightly larger than that in 3-level PWM (3.8%). 3-level PWM is used in our inverter design because of the lower THD.

Persistent Shoot Through Current

A major issue that we faced throughout our simulations was the presence of shoot-through current in our MOSFETs. Figure 62 shows the details of the problem: The red and green waveforms represent the high-side (red) and low-side (green) drive voltages. The voltage inputs were generated from MATLAB to have precisely controlled deadtime (here it is 10µs). Additionally, the rise and fall time of the gate voltages were set to exactly 30ns, in order to ensure that Multisim's Spice algorithms would converge. However, the current through both MOSFETs was seen to spike to unreasonably high levels while turning on the high-side MOSFET. Here, the blue waveform shows the current through the high-side FET, and the purple shows the current through the low-side FET. Note that both currents spike high to about 250A at the same time.

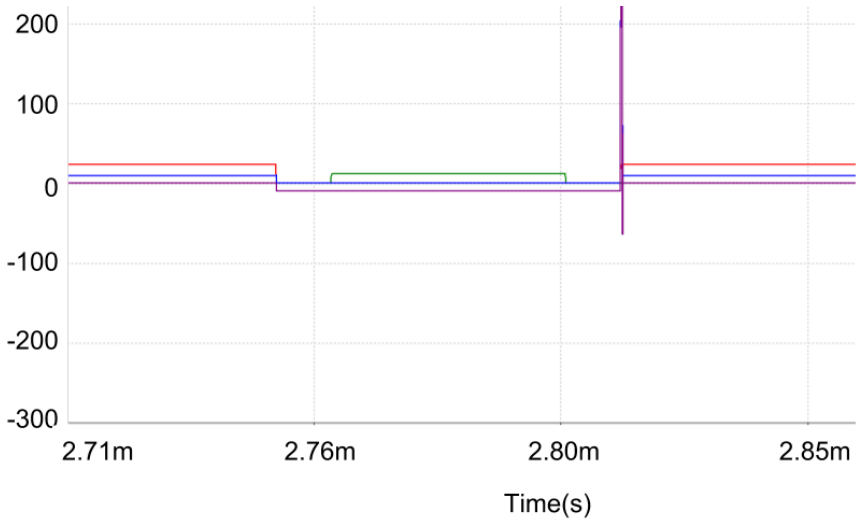


Figure 62: Significant shoot through current despite long deadtime (from simulation)

We tried several approaches to solve this problem, but were unsuccessful at eliminating it. Our approaches included:

1. Adjusting deadtime (up to 30µs) and risetime (up to 30µs) of gate drive voltages
2. Adding snubber circuits around MOSFETs
3. Eliminating the output filter and using a purely resistive load
4. Adding gate resistors for all MOSFETs

In order to continue progress on our design, we decided to move forward with our project despite these issues, in the hopes that differences between the actual design and our simulation would prevent this from occurring. We believed that these issues were artifacts of the IRL2703 simulation model, and that they would not be present in a physical design.

5.2 MOSFET Driver Simulations

Although the UCC27201 has a PSpice model available online, we were unable to successfully create a functional model in Multisim. However, we found that we could

still simulate the device using TI's TINA simulation software. The initial test schematic, using ideal switches in place of MOSFETs, is shown in Figure 63.

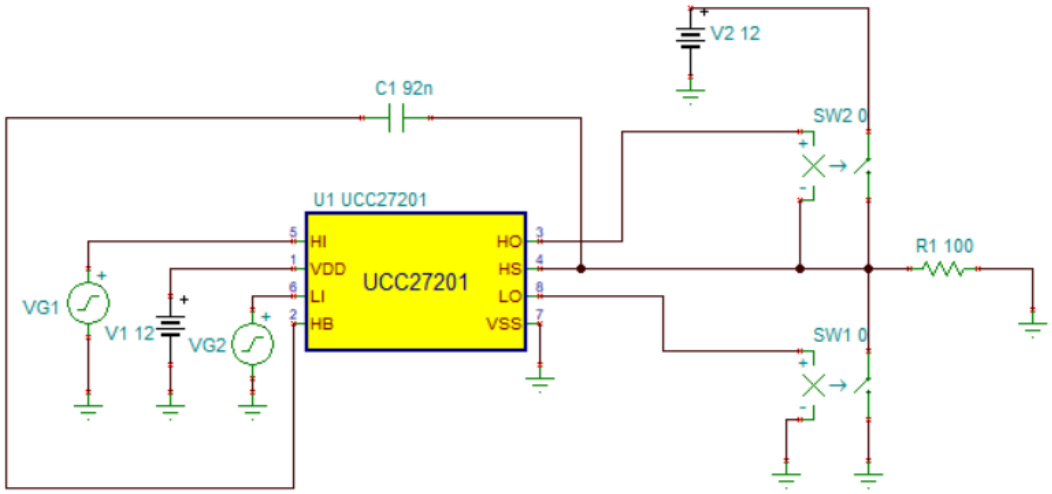


Figure 63: Ideal switch simulation circuit for UCC27201

The HI (high input) and LI (low input) pins are the high and low side input signals. In the real inverter, these signals are to be generated by the microcontroller, but in the simulation in Figure 63 above they are generated by two square wave sources. HS (high source) is meant to be connected to the high side MOSFET source, and HO (high output) and LO (low output) drive the respective gates of the two transistors. The capacitor C1 is the bootstrap capacitor, which is used to generate the high voltage required to drive the gate of the the high side MOSFET. TI did not provide any application notes on bootstrap capacitor selection, so the value of 92nF was chosen using a quick bootstrap capacitor calculator provided by Silicon Labs [36]. This calculator used several criteria, including the switching frequency (24kHz), total gate charge of the IRL2703 (18nC at VG $>$ 10V), minimum and maximum duty cycles (1% and 99%), and allowable capacitor ripple voltage (default is 5%).

A simulation of the high side output is shown in Figure 64. Three waveforms are shown here, which are the high-side input voltage (3.3V square wave), the load voltage (which is also the source voltage for the high side switch), and the high-side gate voltage (24V square wave). Note that the gate voltage magnitude is equal to sum of the power supply voltage for the driver and the battery voltage applied to the half bridge (both are 12V, so sum is 24V). Also note that V_{GS} of the high side switch is equal to 12V, which is the driver input voltage.

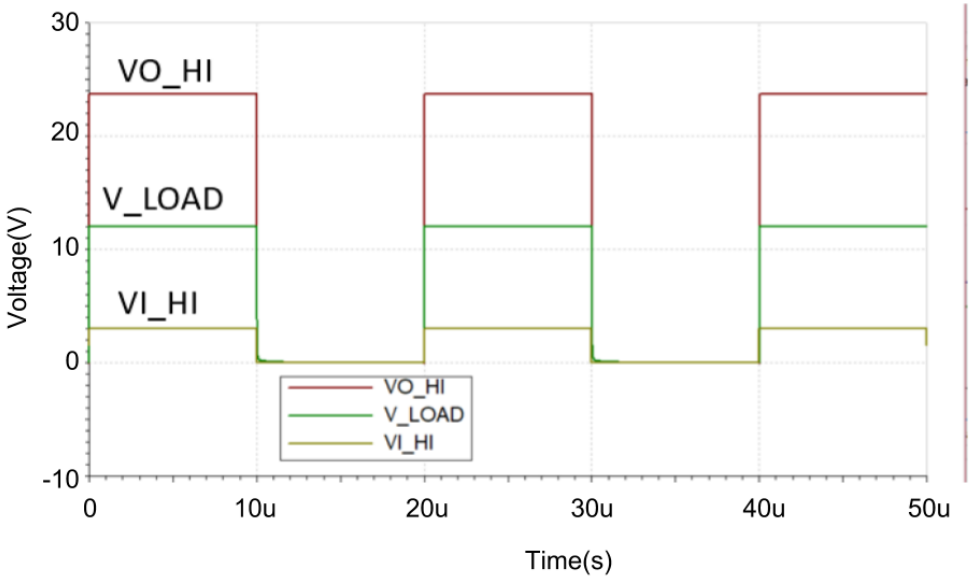


Figure 64: High side output voltage with ideal switches (from simulation)

The next simulation used MOSFETs in place of the ideal switches. Figure 65 shows that the voltage between the FETs (load voltage/high side source/low side drain) varies just as it should - when the high side input goes high, the high side output goes high and the load voltage rises. Similarly, when the low side input is high, the load is shorted to ground and the load voltage is 0V. This confirms the basic operation of the driver is as expected.

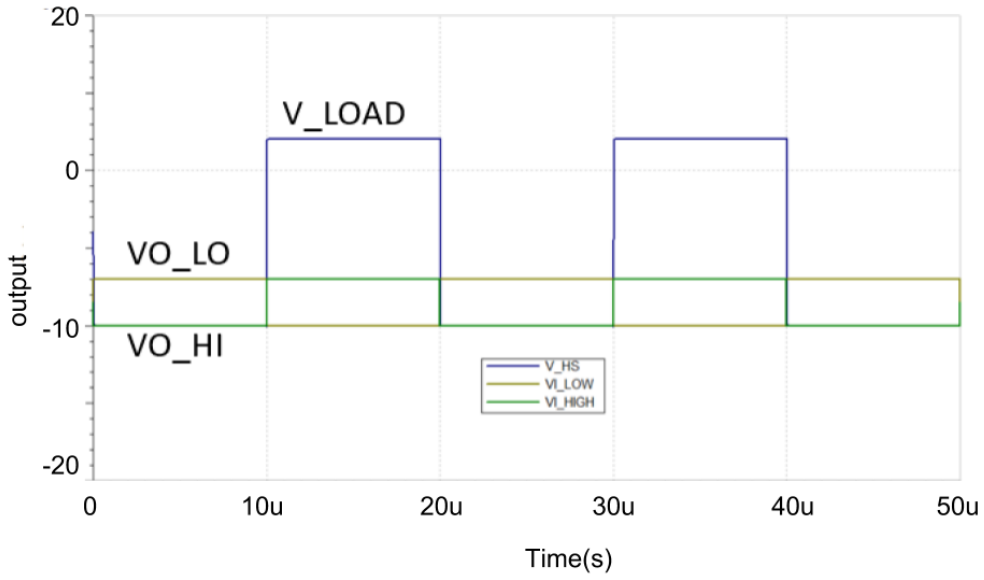


Figure 65: Load voltage with high and low side switches (from simulation)

The next simulation was to confirm that the value of the bootstrap capacitor was acceptable. The high side drive voltage is shown in Figure 66. The voltage probe showed a voltage of 23.5V for the first peak, which eventually drops to about 22.9V after 200us, and stabilizes at that value. This stable voltage represents a drop of only about 1V from the maximum possible, and is stable, so it is acceptable.

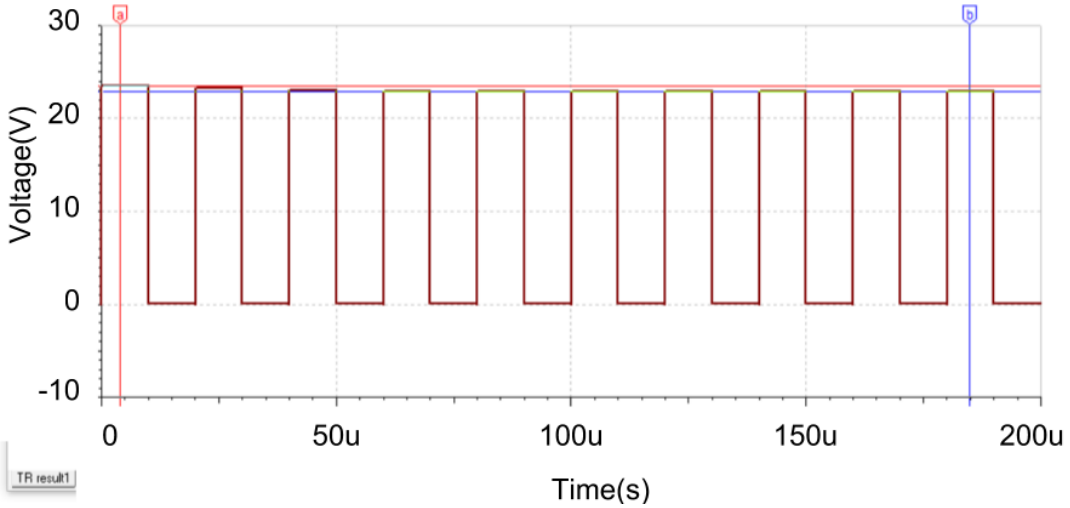


Figure 66: High side gate voltage (from simulation)

Another simulation was completed to check the input current to the gate of each MOSFET. Figure 67 shows the current sourced by the driver into the high side MOSFET (top curve) and the current sinked by the driver from the gate of the low side MOSFET (lower curve). As is shown, the maximum current that is sourced is 1.19A, and the maximum current that is sinked is 1.35A. The current is non-zero for about 500ns. This is less than our expected 1.75A average current, but well within the range of the UCC2720. The current duration is much longer than the specified rise time of the MOSFETs of about 8ns.

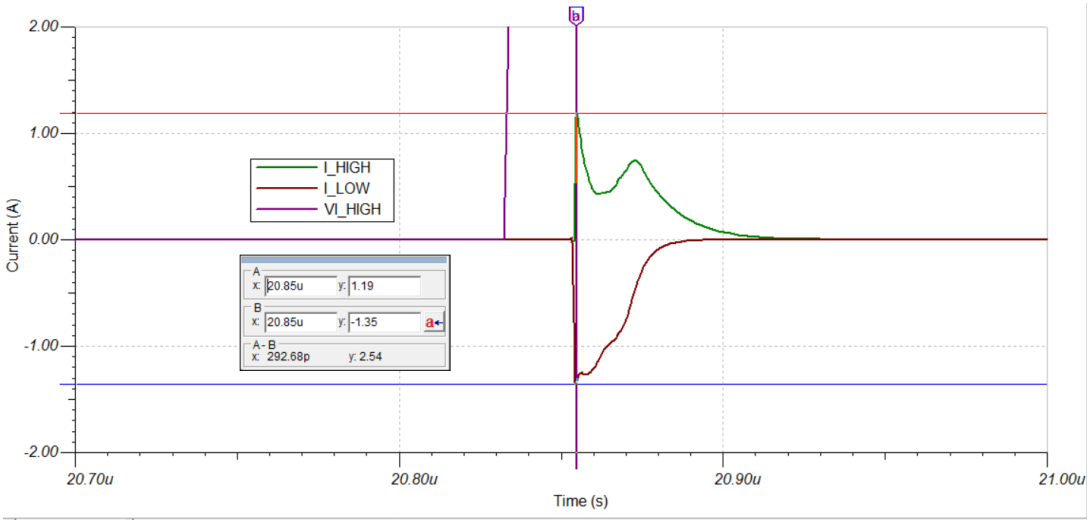


Figure 67: Simulated MOSFET gate current from driver from simulation

5.3 Transformer Simulations

After selecting our transformer, we used information given by Triad Magnetics to model and simulate the transformer's performance, as it had not been fully tested when used as a step-up. Our goals were to model the relationship of input and output voltage while using the transformer in reverse and to understand how much magnetizing current would be required to use the transformer in reverse. The transformer model is described in Table 16.

Field	Value
Primary turns	235
Secondary turns	2 x 7 (center tapped)
Primary resistance	2Ω
Secondary resistance	2 x 6mΩ
Inductance of primary (calculated from magnetizing current)	2.122H
Leakage inductance	0H

Table 16: Transformer Model

The following simulations were generated modeling the input and output current and voltage. The input and output voltages are shown in Figure 68, with a closer look at the input voltage shown in Figure 69. Note that the input voltage (the PWM waveform) does not have a constant magnitude for either its high or low pulses. Note also that there is a phase shift of about $(2\pi)/10$ radians between the two voltages. The currents are shown in Figure 70 - there is no phase shift between the currents. Thus, the input voltage and current are out of phase, while the output voltage and current are in phase. Note also that for this simulation, an input voltage of 12V was used, with PWM chosen to generate a $12V_{peak}$ sine wave for the input of the transformer. It was expected in this situation that the load voltage would be somewhat greater than our $115V_{RMS}$ spec - this will be corrected with feedback-based duty cycle adjustments in our actual device.

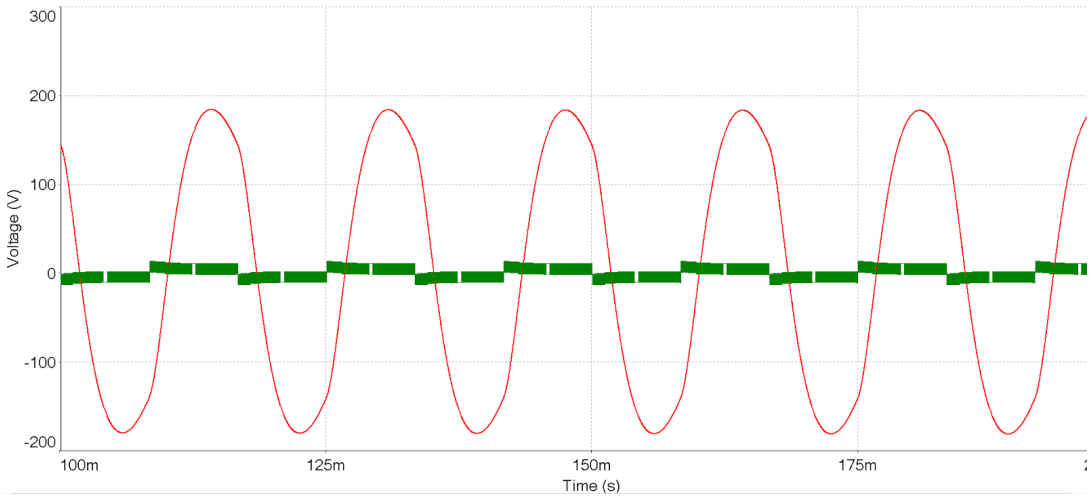


Figure 68: Input (small PWM waveform) and output (large sinusoid) voltage in transformer (from simulation)

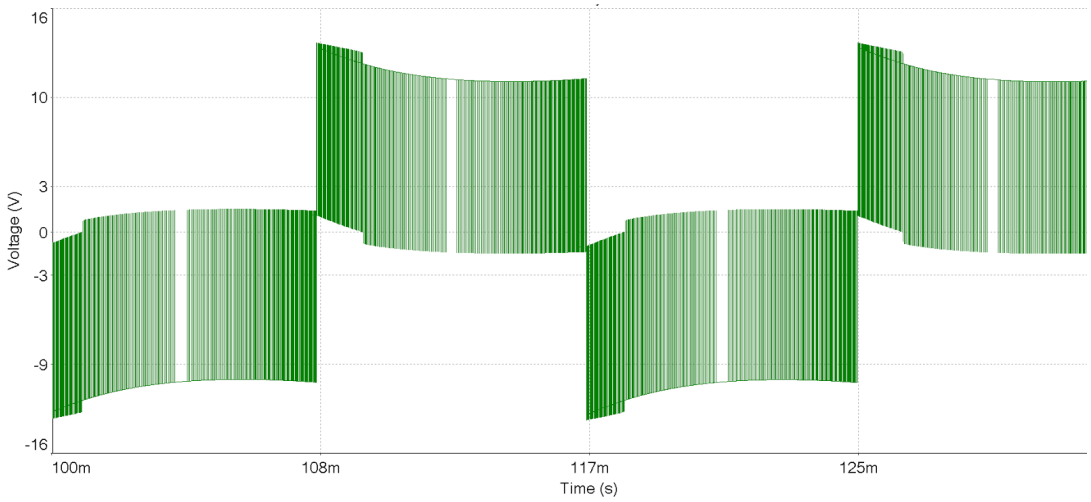


Figure 69: Closer look at PWM input with non-ideal transformer (from simulation)

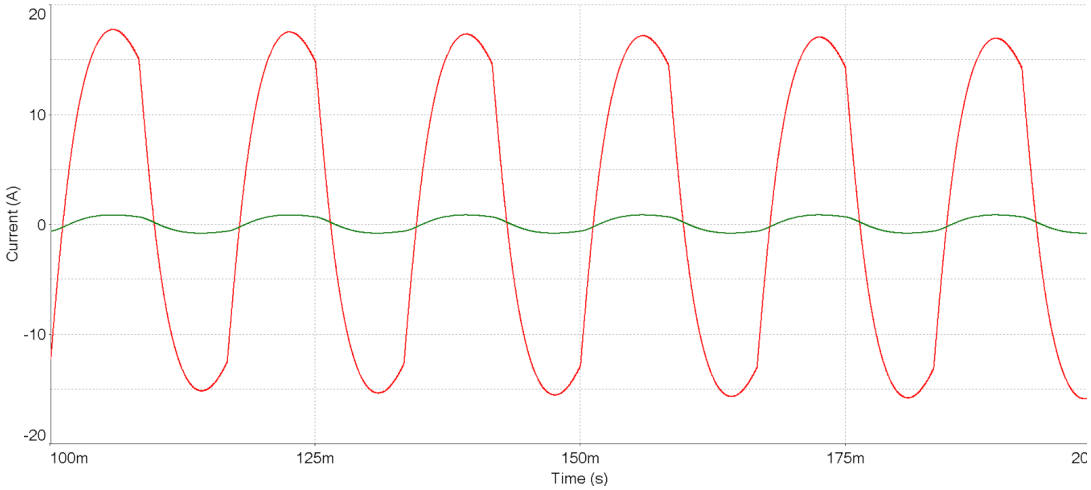


Figure 70: Input (large waveform) and output (small waveform) current in transformer (from simulation)

Table 18 summarizes the results of our transformer simulations.

	Voltage (V_{Peak},V_{RMS})	Current (A_{Peak},A_{RMS})	Phase shift (I lagging V, radians)	Power (W_{Avg})
Input	$12V_{peak}$, $8.5V_{RMS}$	$16.9A_{peak}$, $11.95A_{RMS}$	$2\pi/10$ rad	$82W$
Output	$185V_{peak}$, $131V_{RMS}$	$0.825A_{peak}$, $0.5835A_{RMS}$	0 rad	$76W$

Table 17: Transformer simulation model result

With an ideal transformer model, the input current was measured at only $10.25A_{RMS}$. Thus, the transformer is expected to require about $1.7A_{RMS}$ magnetizing current, based on the significantly higher currents measured with the non-ideal transformer. For reference, Triad Magnetics specifies a magnetizing current of only 120mA when using the transformer in its traditional direction.

We also found that the ratio of output voltage to input voltage is $131V/8.5V = 15.4$. This is notably less than the datasheet turns ratio, which for the reversed transformer would give a voltage ratio of $115/6.3 = 18.25$. However, the transformer will still be able to step our minimum input voltage to an acceptable output voltage: with a $10\text{V}_{\text{peak}} = 7\text{V}_{\text{RMS}}$ input, the output will be about $7V * 15.4 = 108\text{V}_{\text{RMS}}$. Though this is below our nominal output of 115V_{RMS}, it is within the 10% specification we have used for our output voltage, and could be increased by overmodulating the H-Bridge. From these simulations, we can conclude that the transformer will be an acceptable component for our system if its real behavior matches its modeled behavior.

5.4 Input Current Simulations and Resulting Design Changes

After designing our initial schematic, we realized (due to feedback from the NECAM-SID sponsor meeting) that the input current to our circuit would be significantly non-constant, having ripple at both 60Hz and at our switching frequency (nominally 2.4kHz), and potentially negative current. It is unreasonable to expect that a lead acid battery (which is what our project uses) would be able to supply high frequency ripple current and large ripple amplitudes. Thus, it became obvious that a significant capacitance would need to be placed in parallel with the battery in order to source/sink the current ripple, while the battery would provide the average current.

The selection of the input filter relied heavily on simulation. The input current to the H-Bridge from the battery (without filtering) is the pulsed waveform shown in Figure 71. This simulation used a 60W load and 660Hz switching frequency. Note that the current spikes to 20A and -18A. The square wave is the gate voltage of the 60Hz side of the H-Bridge. When the 60Hz side of the circuit is switched from one position to another, the voltage across the load will change as a result. However, the current through the large inductor in our output low pass filter cannot instantaneously change - thus, although the voltage is switched, the current flows in the same direction (through the load) as before the switches were changed. To the battery, however, the direction of this current is reversed.

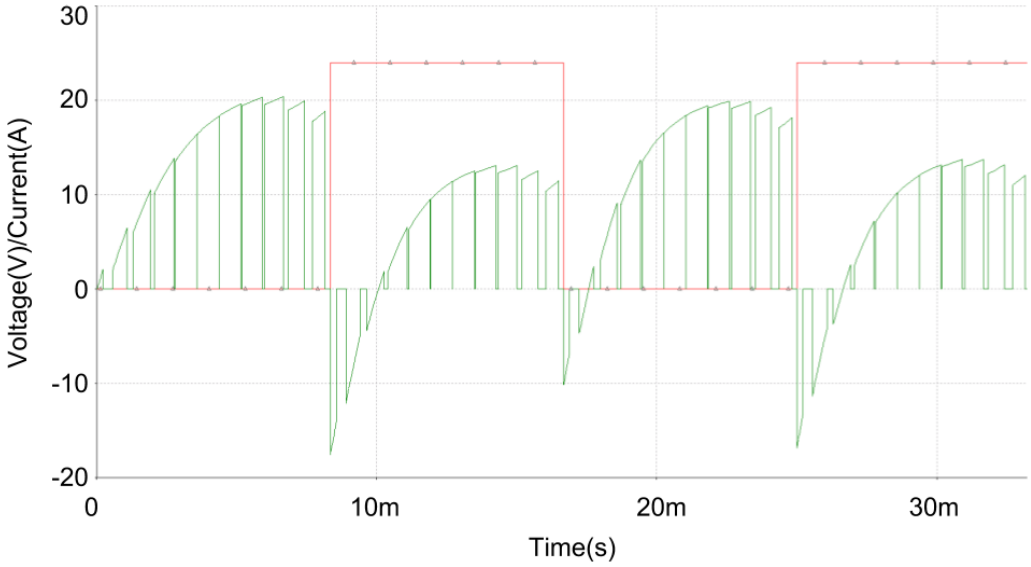


Figure 71: Battery current without input current filtering at 660Hz switching frequency (from simulation)

This same current is shown in Figure 72 with a switching frequency of 24kHz.

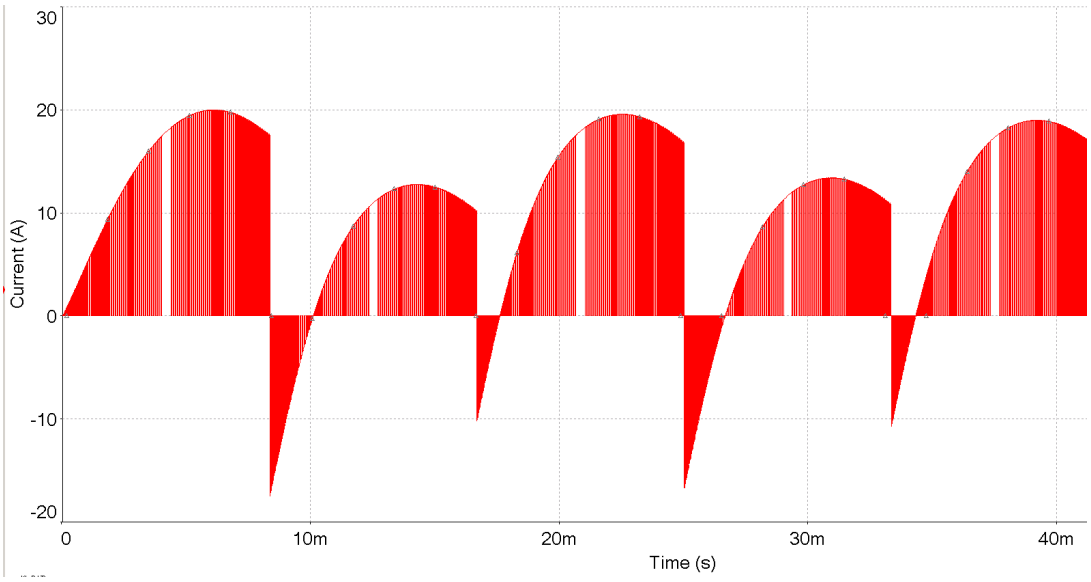


Figure 72: Battery current without input current filtering at 24kHz switching frequency (from simulation)

The simulation circuit used for all of the simulations in this section is shown in Figure 73. To completely eliminate shoot through current and ringing, the circuit uses ideal switches and piecewise linear voltage sources with PWM imported from MATLAB. The MATLAB PWM was configured to have 100ns of deadtime between the turn-off and turn-on of switches in the same branch. Here, unlike in previous simulations, the ideal voltage source was replaced by an ideal battery in series with a small resistance. The resistance was chosen as 11mΩ based on the battery that we purchased for our project. The input capacitance is also shown, with each capacitor's DC resistance simulated using a series resistance. An input inductance is shown as well. Note that for the above simulations, the input capacitors and inductor were removed.

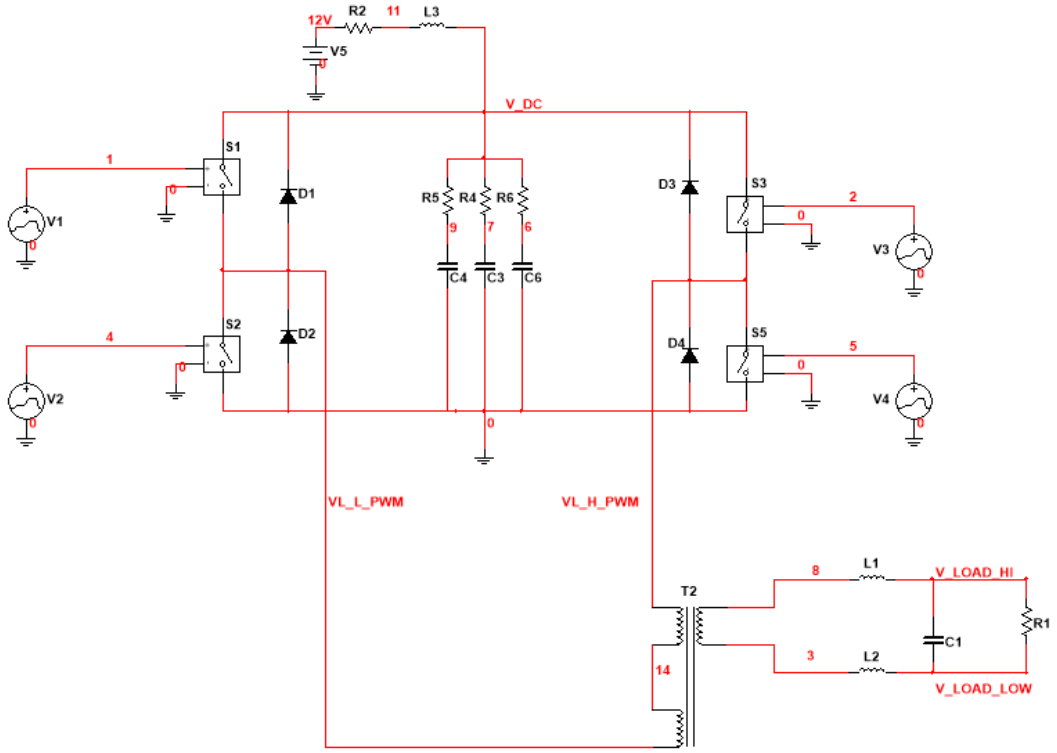


Figure 73: Battery current simulation circuit

After discussing the capabilities of our chosen battery (the UPG UB12500) with UPG tech support, it was determined that a 60Hz ripple current of about $1A_{pk\text{-}pk}$ from the battery would be acceptable. It was also determined that the high frequency switching current ripple should be almost entirely handled by the capacitors. The required capacitance was calculated based on Figure 74, which shows in the green shaded sections the charge required to be supplied by the capacitor bank during each cycle of the 60Hz sine wave. This assumes that the battery current, shown in red (both as positive and negative, though in reality it is exclusively positive) is a constant 11A, and that the input current to the transformer is a $16A_{Peak}$ sine wave. Both of these values were determined via simulation.

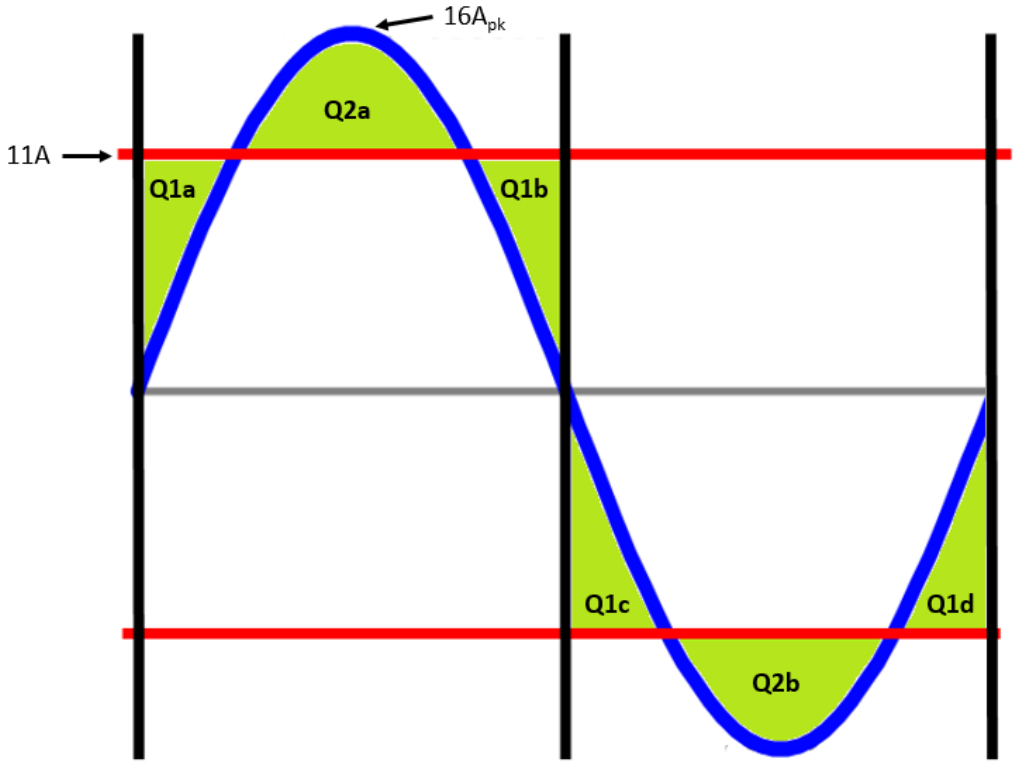


Figure 74: Charge supplied by the capacitor bank during a 60Hz cycle (from simulation)

Integrating to solve for each section showed that the area of Q2 (a or b) is larger than Q1 (a, b, c, or d). Q2 represents a charge of 0.0228C. To calculate a capacitance from this charge, the basic relationship Q = CV is to be used. The voltage for this equation is given by the ripple in voltage permitted across the capacitor, because the maximum ripple will occur when the maximum change in charge is experienced by the cap. Thus, the equation is better phrased as $\Delta Q = C * \Delta V$.

We initially specified a battery ripple of 0.5A, in order to allow for non-idealities in the capacitor and battery that would likely increase the ripple to 1A or more. Assuming that the internal battery voltage is 12V, the external battery voltage due to drop across the battery internal resistance is given by $V = I * R$, where I = 0.5A and R = 11mΩ. This gives V = 5.5mV. Thus, C is calculated as 4.14F, which is very large by typical standards.

A simulation using 4.14F of capacitance showed good results. Figure 75 (top) below is the battery current, which was measured to have a maximum ripple of just under 0.5A. Figure 75 (bottom) shows the capacitor current. The capacitor absorbs both the high and low frequency ripple, and after reaching a steady state has a maximum current of about 8A and a minimum current of -25A.

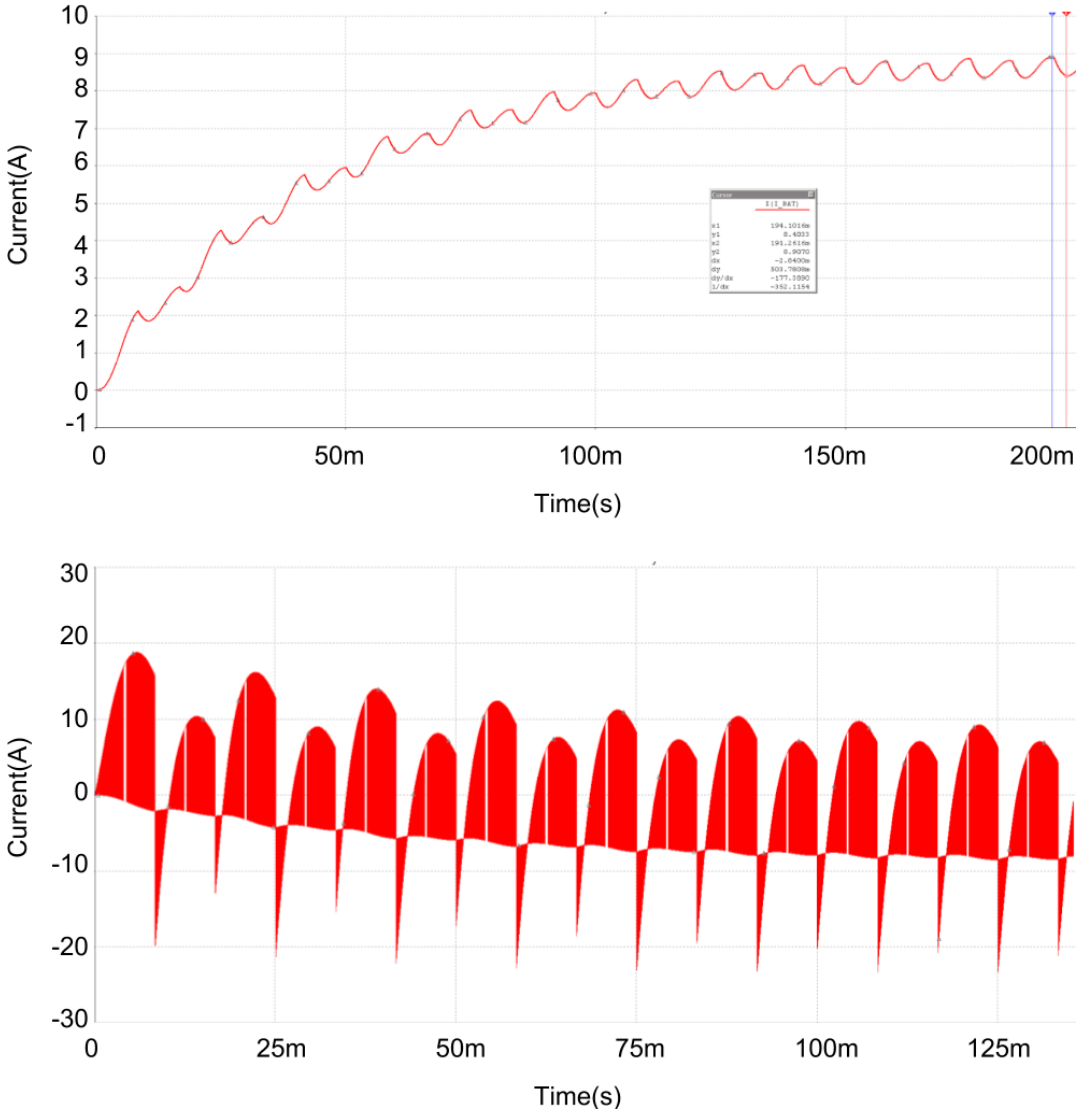


Figure 75: Battery (top) and capacitor (bottom) current with 4.14F of capacitance (from simulation)

Despite the positive result of this simulation, it was discovered upon searching for an appropriate supercapacitor that real capacitors have a series resistance of at least 25mΩ. Moreover, these capacitors come in 2.5-2.7V cells, which must be combined in series to withstand our battery voltage [37]. At a maximum nominal battery voltage of 15V, at least 5 or 6 capacitors must be in series to withstand this voltage. Thus, the ESR of this chain is upwards of 150mΩ. With this level of resistance placed in series with an ideal capacitor, simulations showed that the capacitor could supply little more than 1A total - far from adequate to supply the -25 to +8A shown in the simulation above.

Thus, a new method was tested. Instead of using a very large capacitance with a very large resistance, an inductor was placed in series with the battery to assist in filtering the current and storing energy, and smaller capacitors with higher voltage ratings were to be placed in parallel to decrease effective ESR and increase capacitance.

The simulation result in Figure 76 shows the battery and capacitor currents with a 100μH inductor in series with the battery and 5 x 0.1F aluminum electrolytic capacitors

in parallel with the battery. Each capacitor has only 9mΩ ESR. The battery current had a ripple of just under 1A.

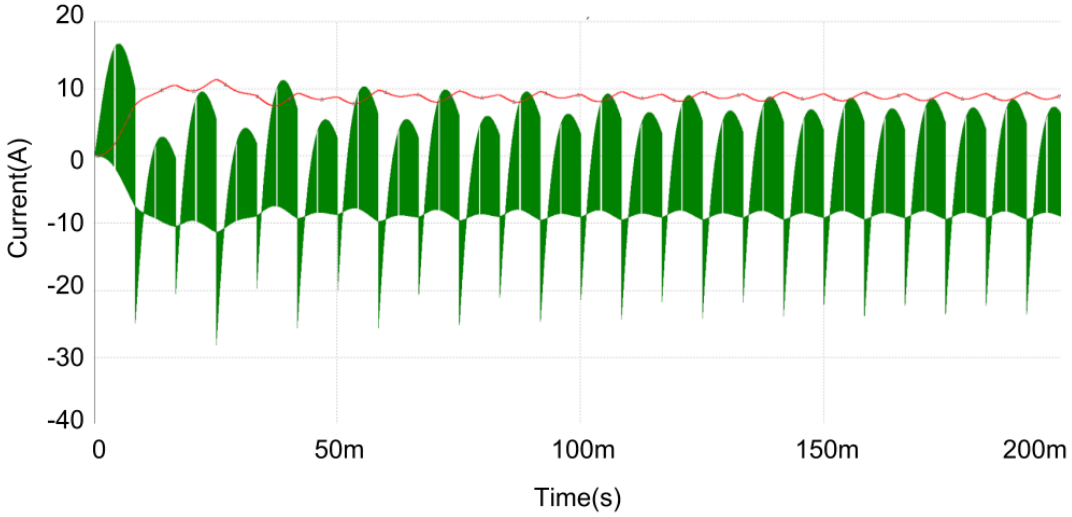


Figure 76: Battery (smoother waveform) and capacitor (pulsed waveform) current with 500mF of input capacitance (from simulation)

The chosen capacitor is the United Chemicon ESMH160. It has a rated voltage of 16V, and each capacitor can supply up to 11.67A ripple current at 120Hz, or 12.6036A at 100kHz. Thus, between 5 capacitors in parallel, the system will be more than adequate to supply the 33A maximum ripple of our system.

5.5 MOSFET Temperature Rise Calculations and Resulting Design Changes

As part of our PCB design, we attempted to estimate the power dissipated by our high-current carrying components and the corresponding temperature rise. The primary components that will carry high currents are the MOSFETs used for our H-Bridge. The power loss was simulated in using the circuit in Figure 77 (identical to the circuit used to simulate the need for input capacitance). Note that an ideal switch is used to eliminate the simulated "shoot through" current seen previously. This switch was given an on-resistance of 40mΩ, which is the value of $R_{DS\text{-}ON}$ specified for the IRL2703. It also has turn-on and turn-off times of 30ns, so that losses generated during switching are not ignored. After this simulation, we found that the IRL2703 was not adequate for our system, and decided to use the NXP PSMN0R9-30YLD instead, which will be discussed later in this section.

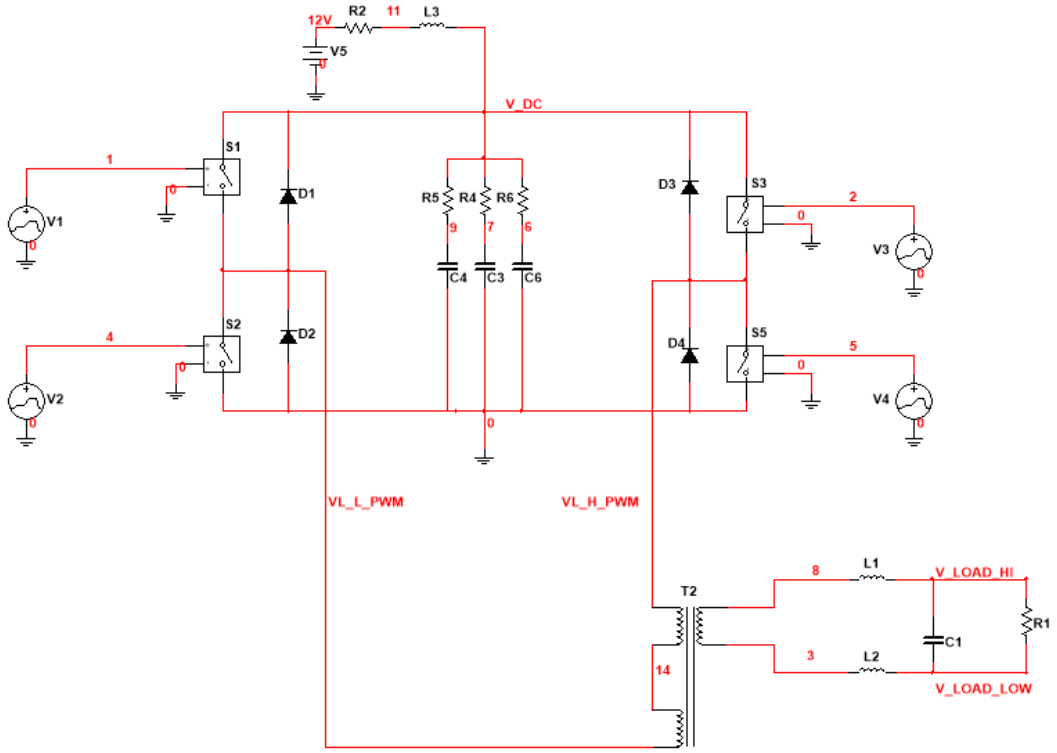


Figure 77: MOSFET power dissipation simulation circuit

The current was measured through S1, which is switched with 24kHz PWM. The voltage is measured across the same switch, and instantaneous power is calculated as the product of instantaneous voltage and current. Average power was calculated in MATLAB, using approximations of the simulated voltage and current given by Multisim. The waveforms generated in Multisim are shown in Figure 78, followed by the MATLAB approximation of current in Figure 79:

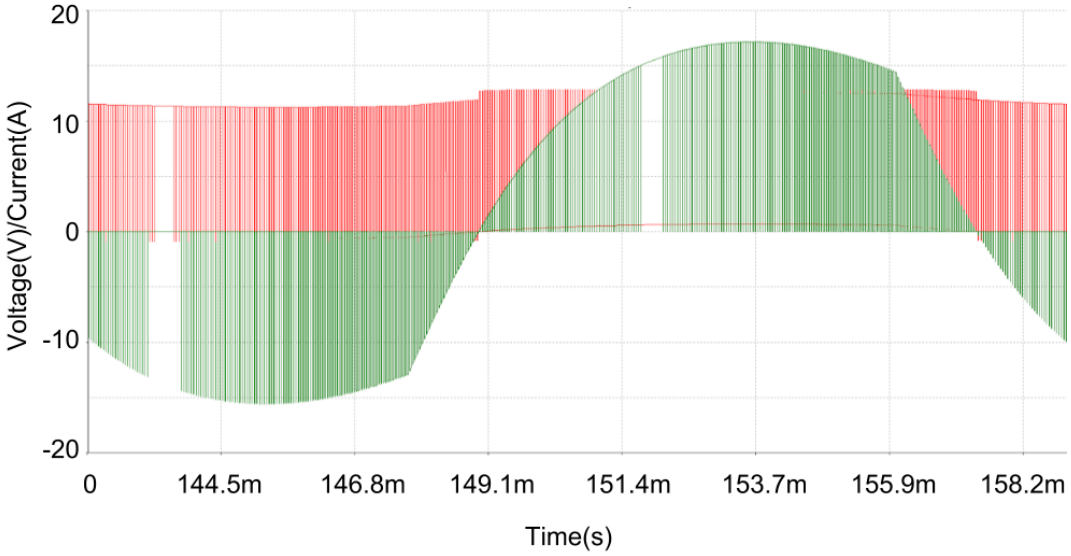


Figure 78: Voltage (red) and current (green approximate sinusoid) through one MOSFET (from simulation)

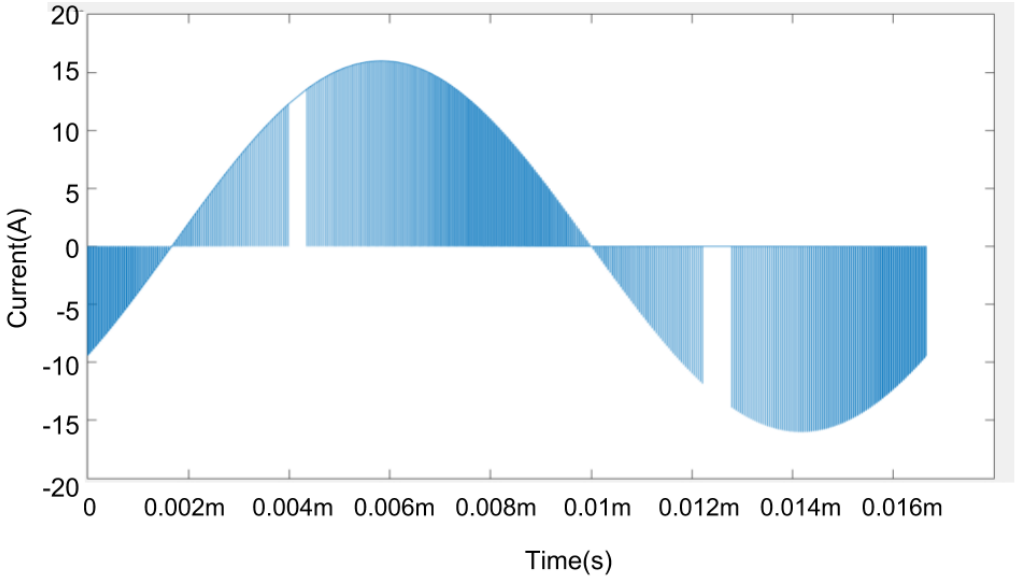


Figure 79: MOSFET current simulated in MATLAB

The instantaneous power was found in MATLAB using $P = I^2 * R_{on}$, and the average power for one MOSFET was found to be 2.56W. The IRL2703 datasheet specifies several thermal resistance values, listed in Table 18:

Thermal Resistance Parameter	Typical Value (°C/W)	Max Value (°C/W)
Junction-to-case ($R_{\Theta JC}$)	-	3.3
Case-to-sink ($R_{\Theta CS}$)	0.5	-
Junction-to-ambient ($R_{\Theta JA}$)	-	62

Table 18: IRL2703 Thermal Resistance Characteristics

From these parameters, the temperature rise of the junction of the MOSFET can be approximated using equation 9:

$$(T_J - T_A) = P_{Diss} * R_{\Theta JA} \tag{9}$$

We can then calculate a first-cut approximation of the maximum temperature rise:

$$(T_J - T_A) = 2.56W * \frac{62°C}{W} = 158.72°C$$

This approximation is crude and unreliable (and likely an overestimate) for several reasons. First, it ignores the natural heat-sinking ability of the connection between the pins and the PCB. Second, $R_{\Theta JA}$ is measured in a very specific lab environment, and is meant mainly to be used as a comparison between different chip case types. Third, $R_{\Theta JA}$ varies non-trivially with power dissipation, ambient temperature, etc. However, this estimate does lead us to believe that some amount of heat-sinking would be necessary for this package to be used. At 70°C, which is the maximum ambient temperature typically specified for by consumer devices, a temperature rise of 152°C would raise the junction temperature to over 200°C. Typically, to ensure device reliability and lifespan, circuits are designed to have a junction temperature of no more than 100 or 110°C, meaning that an estimated junction temperature of over 200°C is unacceptable.

The TO-220 is a vertical through-hole package, which has many available heatsinks or can use the PCB for heat-sinking if it is placed with the metal part of its case on a copper pad on the PCB. We can quickly check if this is feasible for controlling the device temperature by using the following extension of Equation 9:

$$(T_J - T_A) = P_{Diss} * (R_{\Theta Junction-Case} + R_{\Theta Case-Sink} + R_{\Theta Sink-Ambient}) \tag{10}$$

For a first cut calculation, we assume $R_{\theta \text{Sink-Ambient}}$ will be relatively small, and approximate it as 0°C/W. When picking a heatsink or determining the size of a heatsink pad (on the PCB), this value may become significantly greater than 0. However, by approximating it as 0, we can confirm that using a heatsink to lower the temperature dissipation of this device to reasonable levels is feasible:

$$(T_J - T_A) = 2.56W * (3.3\frac{^\circ C}{W} + 0.5\frac{^\circ C}{W} + 0\frac{^\circ C}{W}) = 9.73^\circ C$$

Although a suitable heatsink could be found, the power dissipation of the MOSFET is also problematic from an efficiency standpoint. If each MOSFET dissipates 2.56W on average, the H-Bridge will dissipate over 10W total. If this were the only source of losses, the efficiency of the inverter could be approximated as shown below:

$$E_{inverter} = \frac{P_{load}}{P_{load} + P_{losses}} * 100\% \tag{11}$$

$$E_{inverter} = \frac{60W}{60W + 10W} * 100\% = 85.7\%$$

Given that losses will occur in the input filter, transformer, output filter, and elsewhere, it is unacceptable for such significant losses to occur in the H-Bridge alone. Thus, it was determined that the MOSFETs in the H-Bridge should be replaced with a strong emphasis on minimizing $R_{\text{DS-ON}}$. The chosen MOSFET was the NXP PSMN0R9-30YLD. It is a high-power MOSFET, capable of handling 291W of power dissipation, with an $R_{\text{DS-ON}}$ of only 0.87mΩ. Its major drawbacks compared to the International Rectifier IRL2703 are longer switching times, higher total gate charge, and higher cost. There is also no simulation model availble for the new MOSFET. A comparison of the two is shown in Table 19.

Value	IRL2703	PSMN0R9
V (R_{DSS})	30V	30V
I (R_{DS})	24A	300A
Power Dissipation	45W	291W
R ($R_{\text{DS-ON}}$)	40mΩ	0.87mΩ
Turn on time	8.5ns	38.1ns
Turn off time	12ns	63ns
Package	TO-220 (Thru-Hole)	Power-SO8 (SMT)
Thermal Resistance - Junction to ambient	62C/W	50C/W (with chip mounted on 1x2in copper pad), 125C/W with no copper pad
Cost	$0.95	$1.86

Table 19: Comparison of IRL2703 and PSMN0R9 MOSFETs

Given the lower $R_{DS\text{-}ON}$ of the PSMN0R9, the average power dissipation per MOSFET can be estimated using the previous simulation results and the ratio of resistance between the two FETs:

$$P_{new-FET} = \frac{0.87m\Omega}{40m\Omega} \times 2.56W = 56.6mW$$

With this power dissipation, efficiency (ignoring other losses) can be re-estimated:

$$\eta_{inverter} = \frac{60W}{60W + 0.056W} \times 100\% = 99.9\%$$

This estimate does not account for higher switching losses with the new FET, which switches more slowly. It also does not account for other MOSFET non-idealities that will likely cause losses in the FETs. However, it allows us to estimate the worst case thermal dissipation of the MOSFET using the junction to ambient thermal resistance:
(TJ - TA) = 0.056 * 125 = 7 °C

Given this low temperature rise, it is unlikely that an external heatsink will be necessary. The PCB layout should still be designed with a copper heatsink pad for the MOSFET to ensure it stays cool and to minimize total heat dissipation in the inverter, which will raise its ambient temperature.

6 Implementation

After completing simulation, we tested some parts of our circuit on a breadboard to ensure basic functionality, then designed a PCB. After simulation, we made a few changes to the circuit, which included adding five decoupling capacitors, replacing the H-Bridge MOSFETs with lower power dissipation MOSFETs, increasing bootstrap capacitor values for the driver, and adding gate resistors and gate-source resistors for MOSFETs. The final schematic that we used to design the PCB is shown in Figure 80 below. The microcontroller, transformer, SMPS, and filter inductor are off of the PCB.

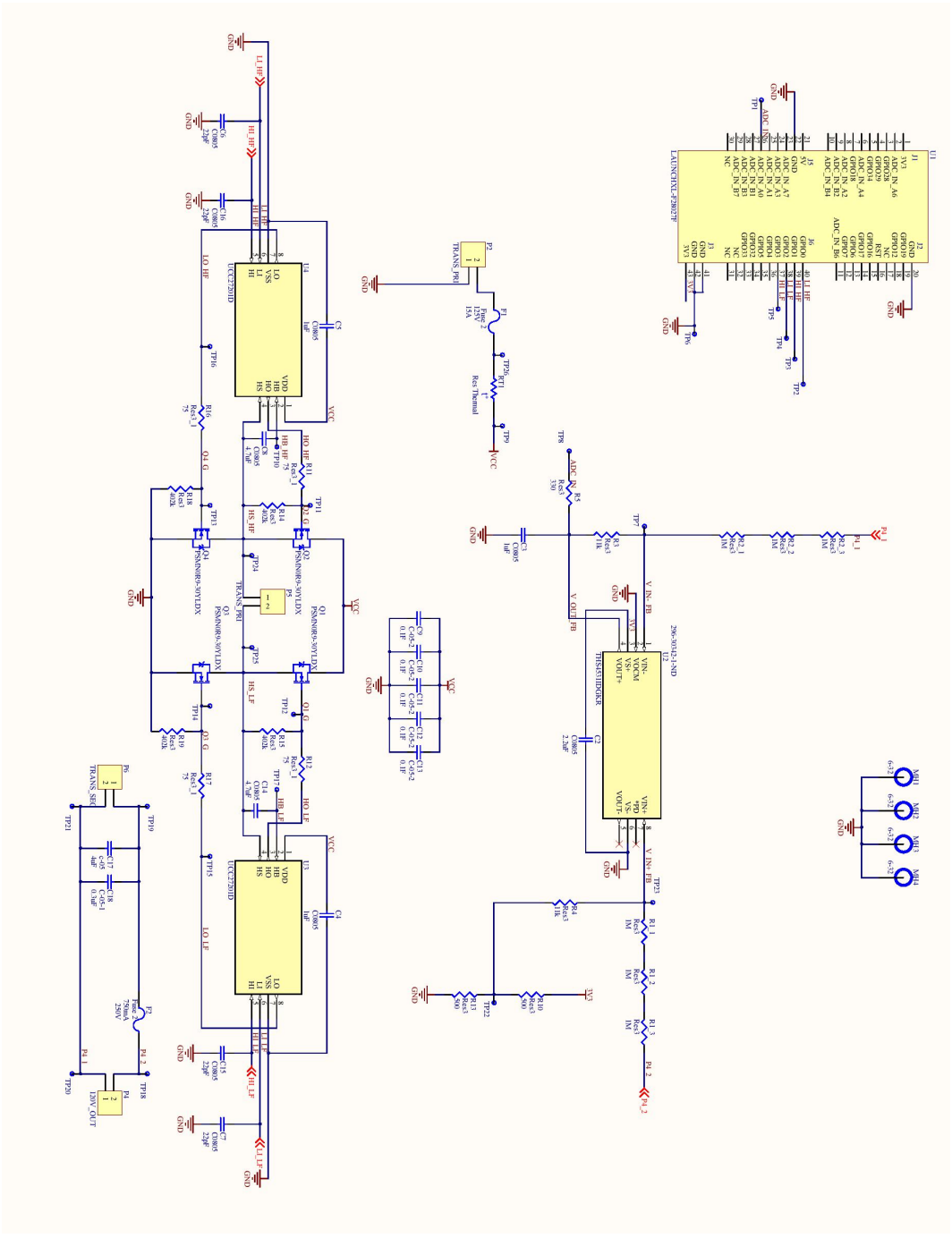


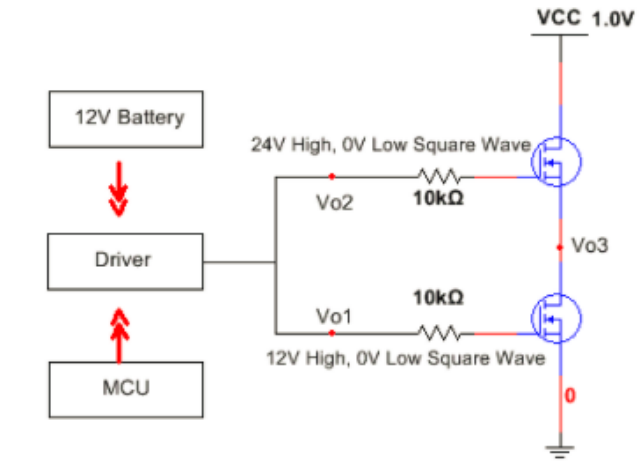
Figure 80: PCB schematic

This section begins with the breadboard tests that we completed. This is followed by a description of the control code we developed. Last is a description of our PCB design and the calculations that went into determining appropriate board layout and trace widths.

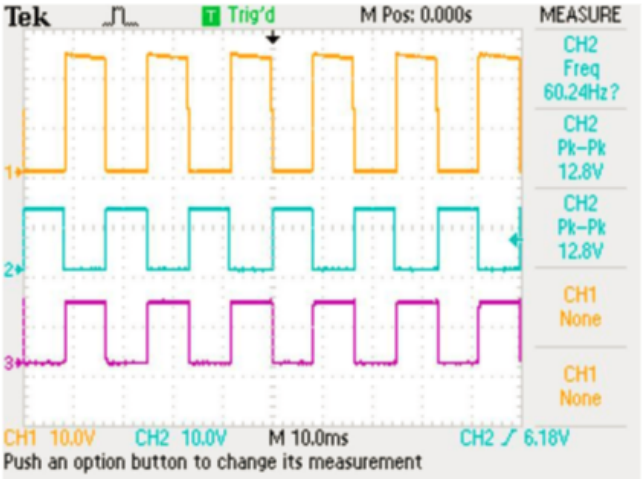
6.1 Circuit Test Plan and Results

Verifying Functionality of MOSFET Drivers

The circuit shown in Figure 81(a) was used to test the UCC27201 MOSFET Driver, the original MOSFET(IRL2703), and microcontroller PWM. We began testing with a 60Hz square wave signal. By connecting the two square wave outputs from the microcontroller (3.3V at 60Hz) and 12V power supply to the driver, we aimed to generate 12V (low side) and 24V (high side) square wave signals from the low and high driver outputs. These two signals are connected to the half bridge, which is made of the two NMOS transistors connected in series. The output can either be connected to ground or to the 12V source when only one of the transistors is on. Thus, the output is a 12V 60Hz square wave. The test results confirmed our expectations, as shown in Figure 81(b).



(a)



(b)

Figure 81: (a) MOSFET driver test circuit and (b) High output (top), Low output (middle), Vo3 (bottom)

In testing the high-side output, it was determined that we need to change the bootstrap capacitor value (we ultimately chose 4.7μF). We tested the circuit with a 92nF bootstrap capacitor first, which was the value that we successfully used in the simulation. However, we found that the capacitor did not hold enough charge to maintain the required voltage, so we used a bigger capacitor instead. Figure 82 is the 92nF bootstrap capacitor test result. Note that the uppermost (blue) signal briefly reaches 22V or so before dropping off as the capacitor voltage drops.

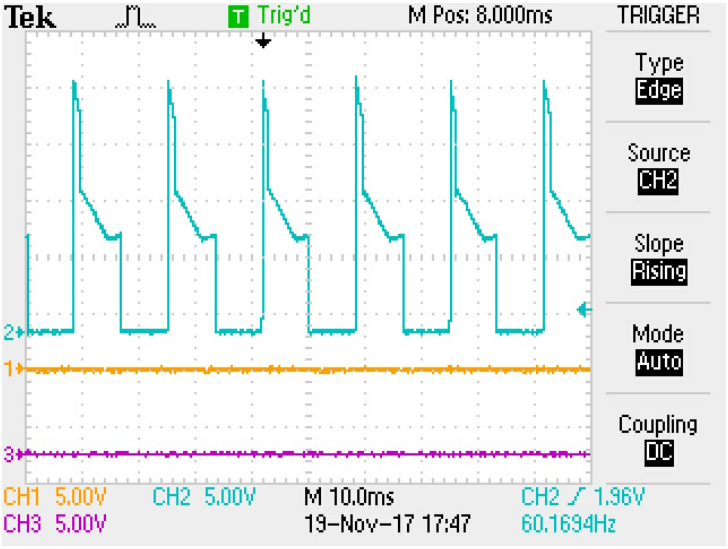


Figure 82: 92nF bootstrap capacitor testing results (blue)

6.2 Microcontroller PWM Generation

PWM generation from the C2000 was shown to be effective. An oscilloscope measurement of the complementary high frequency switching waveforms is shown in Figure 83. Note the characteristic sine wave PWM shape, and that the oscilloscope measured frequency is approximately 24kHz.

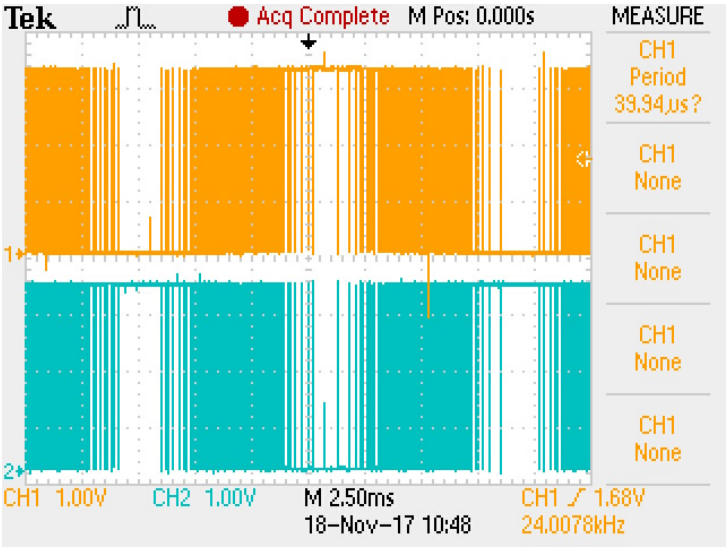


Figure 83: 24kHZ PWM generated from C2000

Figure 84 shows a closer look at the PWM generated at 24kHz. At any time, the first waveform has a low duty cycle when the second waveform has a high duty cycle. Note that the two waveforms are complementary.

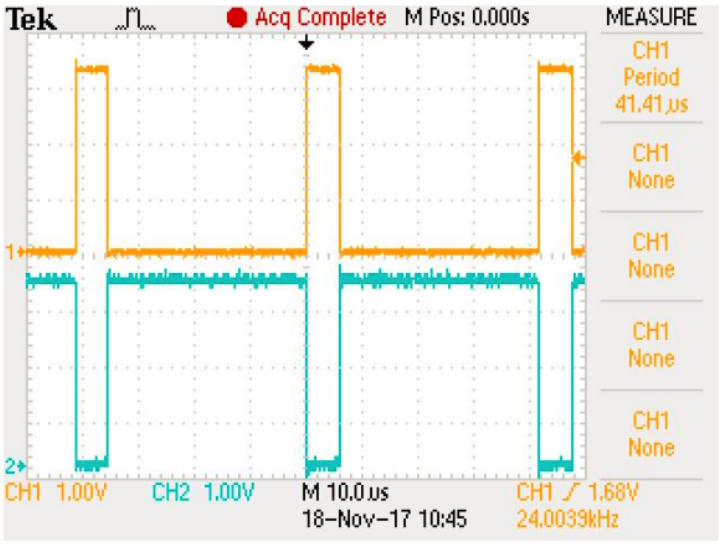


Figure 84: Zoomed in 24kHz PWM generated from C2000

Another key feature of this PWM is the deadtime, which is shown in Figure 85. Here, it was calibrated to be 166ns on the C2000, and was measured at approximately 164ns. It is adjustable down to $\frac{1}{2}$ of 1 clock cycle, or 8.33ns at 60MHz clock frequency.

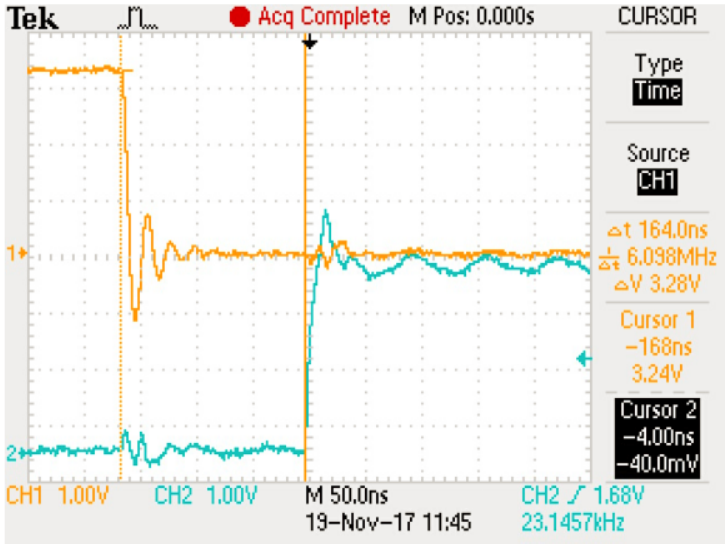


Figure 85: 166ns deadtime on 24kHz PWM from simulation

Initial breadboard tests with PWM frequencies above 60Hz were unsuccessful. This is most likely because of the relatively high inductance connections between different components (especially between the driver and MOSFET).

6.3 Voltage Amplitude Control Feedback Loop

There are 3 main components to be implemented in order to complete the feedback loop. The first component in the loop is the sampling circuit used to scale mains-level voltages ($\pm 170 V_{pk}$ or more) to the 0-3.3V range of the ADC. At this point in the design process, this circuit had been designed in simulation software but was not physically tested until the inverter PCB was assembled. The second main block in the loop is the AC/DC conversion block. Implementation and preliminary testing of this block were completed at this point in the process, because it is built entirely in software on the C2000 MCU, and interfaces with hardware via an Analog to Digital Converter (ADC). The third block is the PID controller, which could not be tested until the MCU was tied to an otherwise fully functioning inverter.

The AC/DC conversion block is based off of our brief exploration into I/Q sampling discussed in the Background section. The sampling process occurs thousands of times per second - the steps taken on a single sample are outlined in the list below. This list is sequential and is meant to occur very quickly - after an ADC sample is taken, the entire sampling and AC/DC conversion sequence must happen before the next sample is taken. The order of steps is as follows:

1. ADC periodically takes a sample of output and triggers an interrupt when the sampling sequence is complete
2. ADC interrupt service routine (ISR) begins to execute
3. ADC ISR accesses new ADC sample and stores as the newest value in a buffer of samples
4. ADC ISR takes an older sample from buffer to act as the current value of "imaginary" (-90° phase shift) value
5. ADC ISR finds approximate magnitude of sampled sine wave from I/Q sampling method

Figure 86 shows how the ADC circular sample buffer is used to create an imaginary waveform. Samples are stored from left to right; when the end of the buffer is reached, the ADC wraps around to the beginning of the buffer. Samples are overwritten some time after they are taken, but only after they have been used and are no longer needed for calculation. The sample used for the imaginary waveform is offset from the current sample by a set number of samples - for example, with a 60Hz waveform and a 48kHz sampling frequency, 48,000 / (60 / 4) = 200 samples are recorded during each 90° portion of the waveform. Thus, the offset used to acquire the imaginary waveform is 200 samples. The buffer is sized accordingly - with a 48kHz sampling frequency it should be at least 200 samples long (and preferably not much longer to reduce RAM usage).

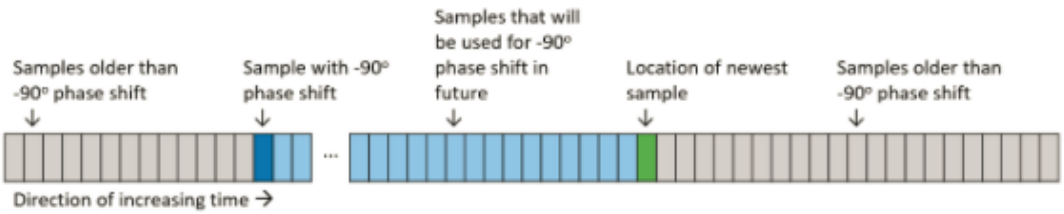


Figure 86: ADC circular sample buffer

6.4 Printed Circuit Board Implementation

We were successful in completing the design of the printed circuit board (PCB) of the circuit on the first rendition. We used Altium® as our design software, which is one of the industry standard PCB design softwares, has accessible tutorials, and which quite a few people on WPI's campus are knowledgeable about and were willing to provide assistance with. Our team received access to a free Altium license from the Gordon Library IT Help Desk. The layout shown in Figure 87 is the circuit built in Altium. It is a four-layer board - blue traces represent those on the bottom layer and red traces on the top layer. The middle layers are a 12V power plane and ground plane.

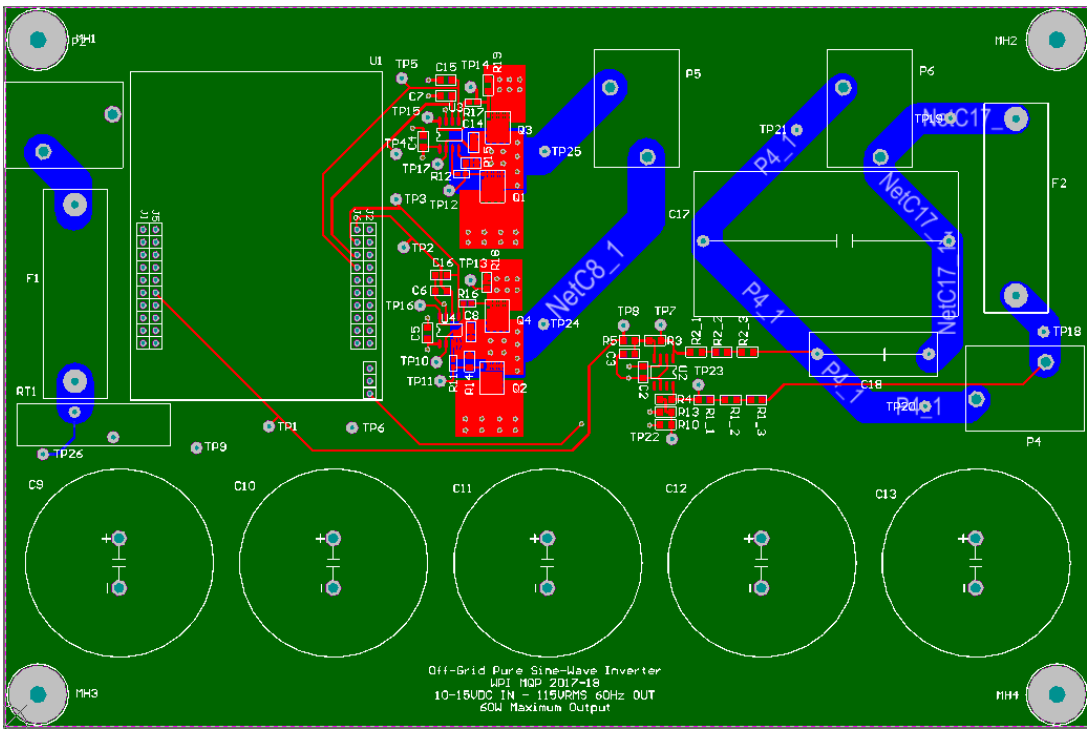


Figure 87: Layout for printed circuit board [218.44 x 144.78 x 2.11 mm]

We decided to make a four-layer PCB in order to connect components with VCC and ground easily, reduce power loss from heavy currents from VCC, and also reduce the noise between analog and digital signals. All components are placed on the top layer of the board. The top layer is also used primarily for low power and sensitive analog/digital signals. The second layer is the GND plane. The third layer is the 12V (VCC) plane, which is connected to the battery (through a fuse and inrush current limiting resistor). The bottom layer is for high power traces and is also used to route some small-signal traces that needed two layers for proper routing.

There are some high current (up to about 10A) traces in our circuit, so we needed to use wide traces to reduce power loss and temperature rise in them. We used a calculator based on IPC-2152, the "Standard for Determining Current-Carrying Capacity In Printed Board Design", to determine our trace widths [38]. Table 20 and Table 21 show the power loss when using 1oz copper and 2oz copper for the 10A path from the battery to the H-Bridge and input filter capacitors (assuming each of these paths is 5 inches long). As we can see from the tables, when current is 10A, using 2oz copper can reduce power loss by about half (compared to 1oz copper) when using the same trace width, so we decided to

use 2oz copper for our PCB. Although 3oz copper has a better performance, the cost of each board ($300) is almost double the price of a 2oz copper PCB ($160), so we chose not to use 3oz copper.

We used 300 mil width for large current traces in our PCB. When traces are 300 mils wide, the power loss is about 0.7W for 1oz copper and 0.3W for 2oz copper. Using 2oz copper at this width gives a good trade-off between maximizing board space while minimizing power loss. The power loss is calculated based on 5 inch long traces, so the actual power loss will be smaller in all of our traces (none of our traces is more than about 2.5 inches long). For traces that are not expected to conduct significant power, we have used 15mil traces.

°C Rise	Width(mils)	Width(mm)	Voltage Drop	Power Loss(W)
2	2997	76.1238	0.008688	0.08688
5	983	24.9862	0.0256	0.256
10	497	12.6238	0.0524	0.524
20	263	6.6802	0.0990	0.990
30	181	4.5974	0.144	1.44
45	129	3.2766	0.202	2.02

Table 20: 10A current with 1oz copper

°C Rise	Width(mils)	Width(mm)	Voltage Drop	Power Loss(W)
2	1653	41.9862	0.00788	0.0788
5	553	14.0462	0.0235	0.235
10	278	7.0612	0.0468	0.468
20	140	3.556	0.0930	0.930
30	97	2.4638	0.134	1.34
45	68	1.7272	0.191	1.91

Table 21: 10A current with 2oz copper

After completing the PCB layout, we chose PCBWay to fabricate the board because it is less expensive and faster than many other manufacturers. Figure 88 is the unpopulated PCB manufactured by PCBWay. The size of the board is 218.44 x 144.78 x 2.11 mm.

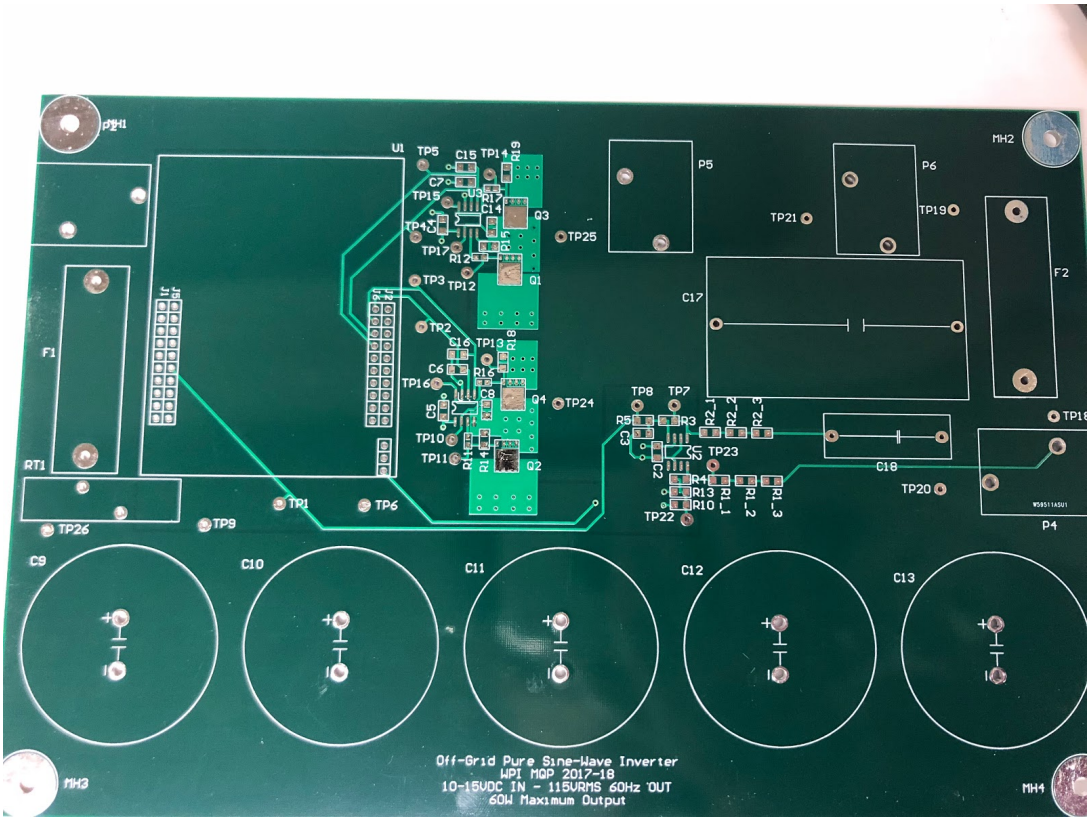


Figure 88: Unpopulated PCB

7 Results

This section will discuss the functionality of our completed PCB. Test results include verification of basic inverter functionality, verification and tuning of the voltage amplitude regulation feedback system, and results of testing the inverter with a variety of loads. This section will also discuss the reliability calculations we performed.

7.1 PCB Assembly and Basic Functionality Testing

Figure 89 is the populated PCB. There are five input capacitors on the top of the board, a microcontroller on the right side of board, an H-Bridge circuit in the middle of the board, and an LC low pass filter on the left side. The battery is connected to the rightmost terminal block, and the leftmost terminal block is connected to the AC load. The two terminal blocks on the bottom of the board connect to the chassis-mount step-up transformer and filter inductor, both of which are located off-board.

Figure 89: Populated PCB

After soldering all the components onto the board, we tested the microcontroller's output signals on the PCB. Figure 90 shows the microcontroller PWM output signals. We used a switching frequency of 2.4KHz instead of the 24KHz switching frequency that we used in the simulation. During testing, we determined that the source voltage of the high side MOSFET in each pair took 3 to 5µs to stabilize after a switch transition. To avoid any possibility of shoot through current in either half-bridge, we used a 15µs deadtime between switching the two MOSFETs. This limited our maximum switching frequency because at higher frequencies, this relatively long deadtime visibly distorted the PWM signals generated by the microcontroller by preventing short pulses from ever

occurring. Thus, 2.4kHz was an approximate maximum for our switching frequency. This did not negatively impact our output waveform because our large output filter has a cutoff frequency of 115Hz, well below our switching frequency.

The four PWM outputs from the C2000 at 2.4kHz are shown in Figure 90. The outputs appear to be correct - the "center" of the PWM channels line up with the midpoint of the square wave channels.

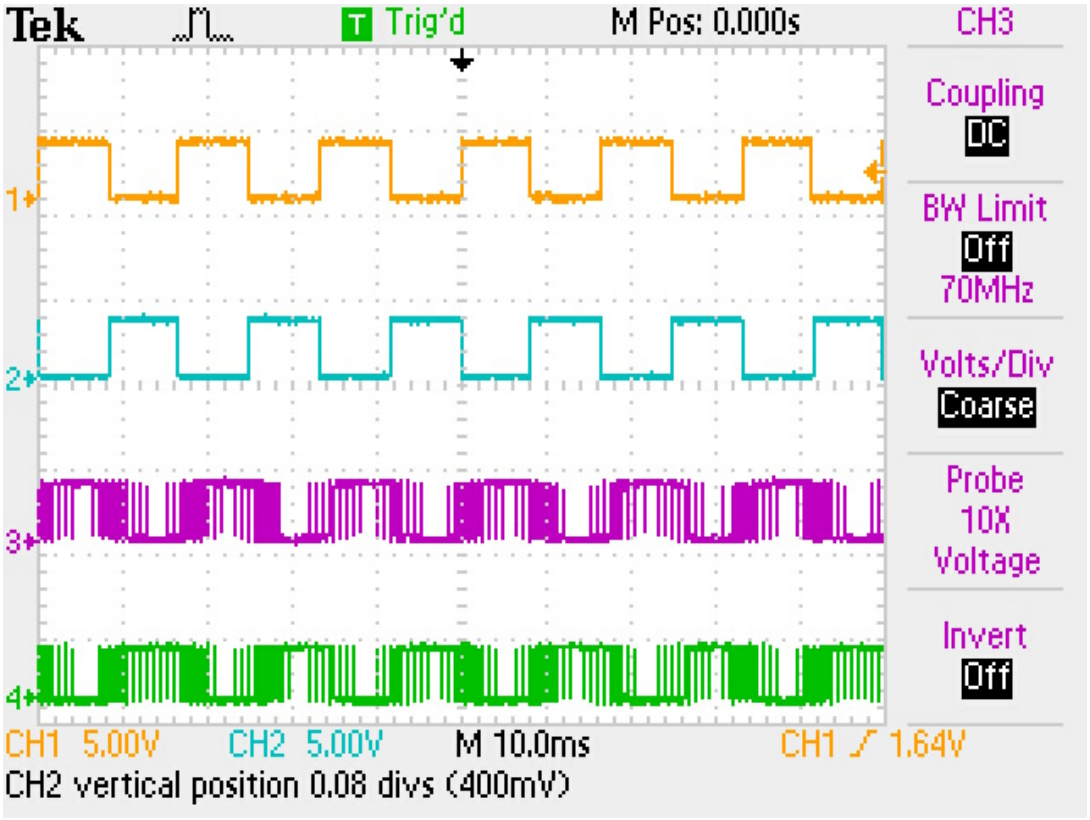


Figure 90: Microcontroller PWM output

Figure 91 shows the measured driver high output (HO), low output (LO) and high-side source (HS) pins. Unlike in our breadboard tests, all of the driver outputs appeared to function properly regardless of switching frequency. Our circuit test set up is shown in Figure 91.

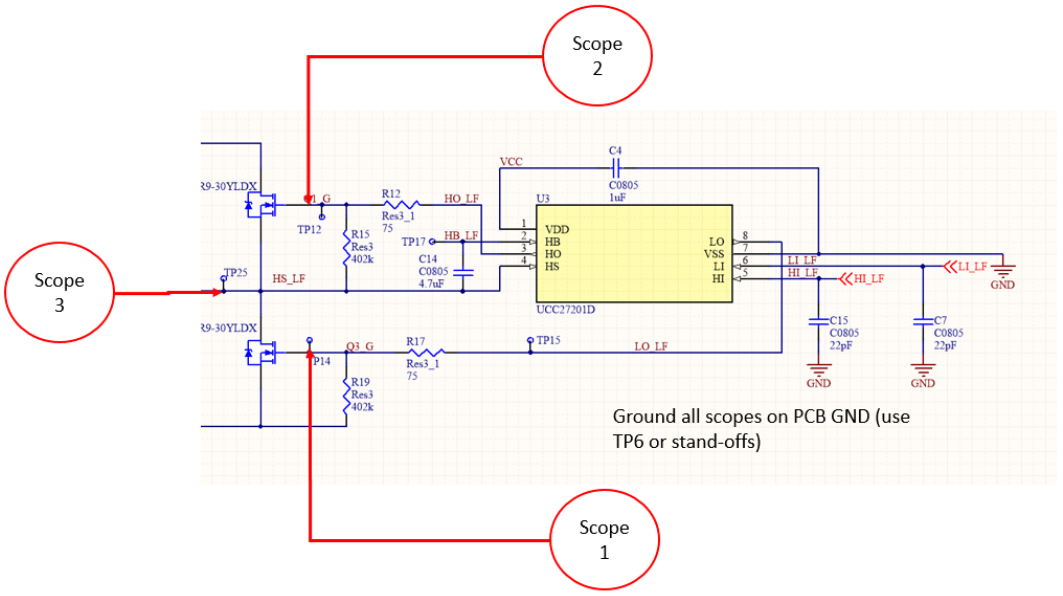
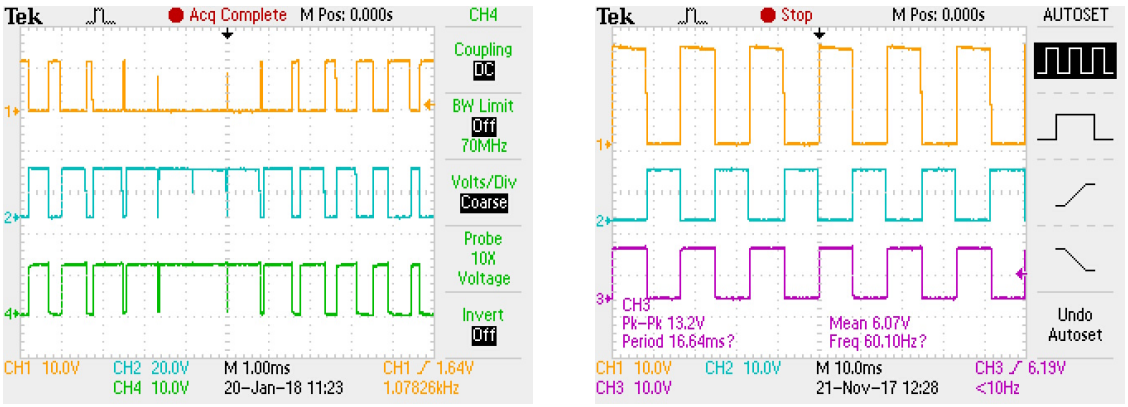


Figure 91: Driver test circuit

The circuit appeared to function correctly, and we could even hear the MOSFETs switching (2.4kHz is within human hearing range). Figure 92(a) shows the output signals from the "high frequency" driver. The uppermost signal is low output (LO), the middle signal is high output (HO), and the bottom-most signal is high-side source (HS). Figure 92(b) shows the output from the 60Hz driver. The uppermost signal is high output, the middle signal is low output, and the bottom-most signal is HS.



(a) (b)

Figure 92: (a) High frequency driver output (b) 60Hz driver output

We found that the MOSFET gate voltage with our given gate resistor has a 3µs rise time, shown in Figure 93. The sharper, yellow waveform is the output measured directly at the driver pin, and the slower, blue waveform is the voltage measured at the MOSFET gate (after the gate resistor). It was decided that deadtime for each transistor should be much longer than this, to ensure that each MOSFET is fully off before the other MOSFET begins to turn on. We settled on a deadtime of approximately 15µs.

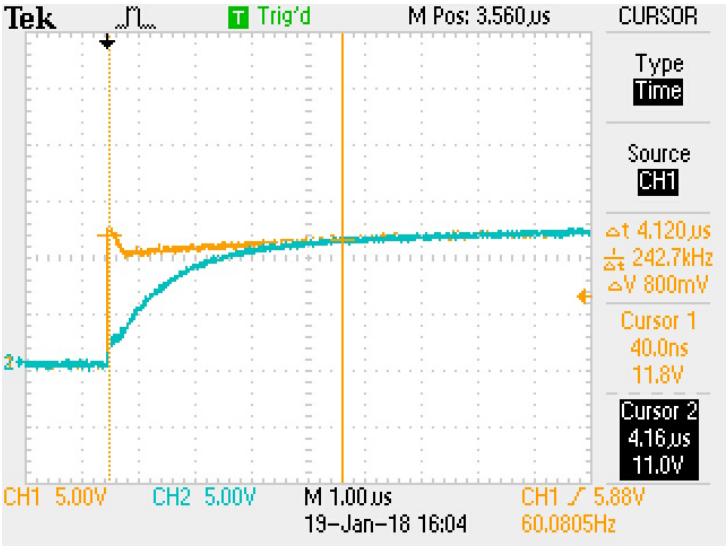


Figure 93: MOSFET gate rise time

The topmost signal in Figure 94 shows the HS voltage between the 60Hz MOSFETs, and the bottom-most signal shows the HS voltage between the high-frequency MOSFETs. The second signal is 3-level PWM after the H-Bridge (i.e. the transformer primary inputs). We used two oscilloscope probes to measure the voltage between the 60Hz MOSFETs and voltage between high-frequency MOSFETs, and used the oscilloscope MATH function to display CH1-CH2, which is the voltage difference between those two points, shown in the middle, red waveform.

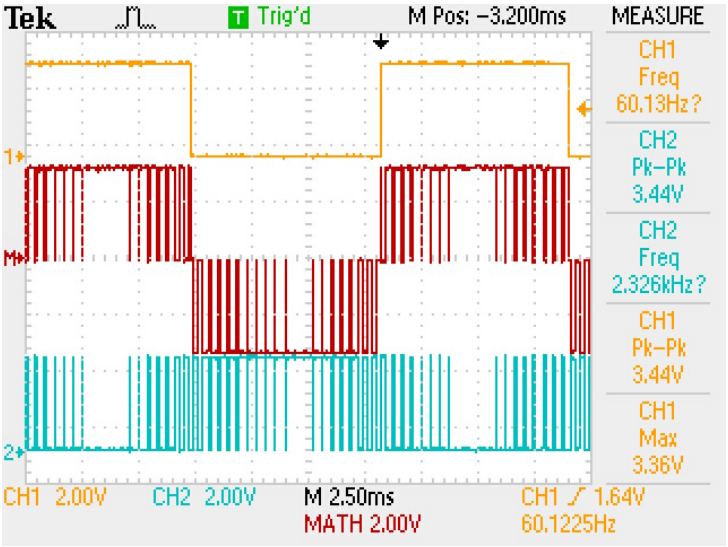


Figure 94: 3-level PWM output

We discovered several small problems in our PCB layout during testing. First, the footprint for the drain pad of our MOSFET was slightly too small, so it was difficult to solder. Second, we ordered the incorrect model of our MOSFET driver. We ordered the UCC27200A, which has CMOS input thresholds while the UCC27201A, which is the correct driver, has TTL-compatible thresholds. The UCC27200A input rising threshold is (at minimum) 5.8V, which is too high for our 3.3V microcontroller to source, so the microcontroller signal to the driver was always low, and we never saw any output from

the driver until we realized our error and switched drivers. The third issue was that the PD_n (active low power down) pin on our feedback circuit op-amp was left floating on the PCB, but needed to be driven high to operate the op-amp normally. We simply soldered a wire from the pin to a 3.3V via on the board, and the op-amp functioned as expected. The last problem we had was a minor issue with our terminal block - it is relatively easy to plug a wired into the terminal block but difficult to remove the cable, so we suggest choosing a terminal block with easier wire removal for a prototype device.

7.2 Testing Circuit with a Purely Resistive Load

Once we generated a 3-level PWM signal as we expected, the next step was to add the transformer, output filter, and load to the circuit, and to see if the circuit could generate a low-THD sine wave. The circuit with transformer, filter and load is shown in Figure 95.

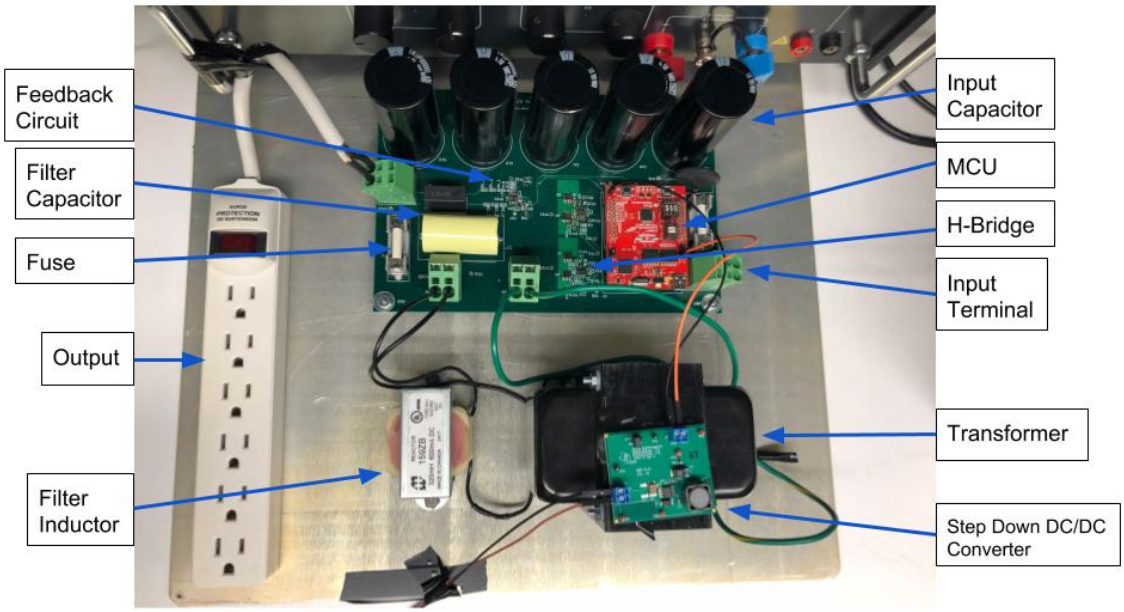


Figure 95: Full inverter circuit - PCB with filter and off-board transformer and inductor

We first tested with a 220Ω power resistor for a load because a 220Ω resistor dissipates 60W at 115V, so it is an ideal load for our inverter. Figure 96 is the voltage across the power resistor - our first revision PCB successfully generated a 60Hz sinewave. The output AC signal was displayed on the scope by using the MATH function CH1-CH2. However, because the feedback circuit was not working at that time, the output voltage is not $115V_{rms}$ exactly. Therefore, a feedback circuit is necessary to keep output voltage stable when input voltage changes.

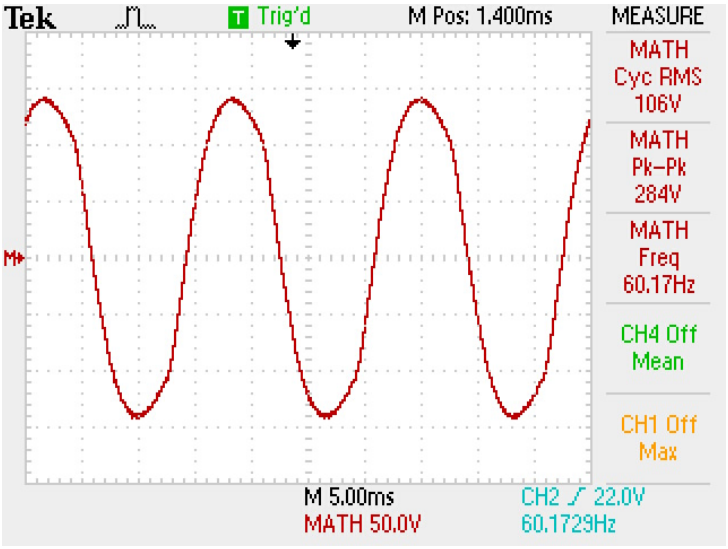


Figure 96: Voltage measured across the power resistor with 9.8V input

A Fourier analysis (FFT) of the inverter output was used to calculate the harmonic content of the waveform. As we can see from Figure 97, the third harmonic (f = 180Hz) contributes most of harmonic distortion, and there is also some distortion from the 5th, 7th, 9th, etc harmonics. This FFT was taken using the Tektronix TDS2000 oscilloscope with "Hanning" FFT windowing. THD is calculated as shown in Equation 12. In this equation, n represents the harmonic number, where $n = 1$ represents the value of the fundamental.

$$THD = \sqrt{\frac{\sum_{n=3}^{n_{max}} {V_n}^2}{{V_1}^2}} \tag{12}$$

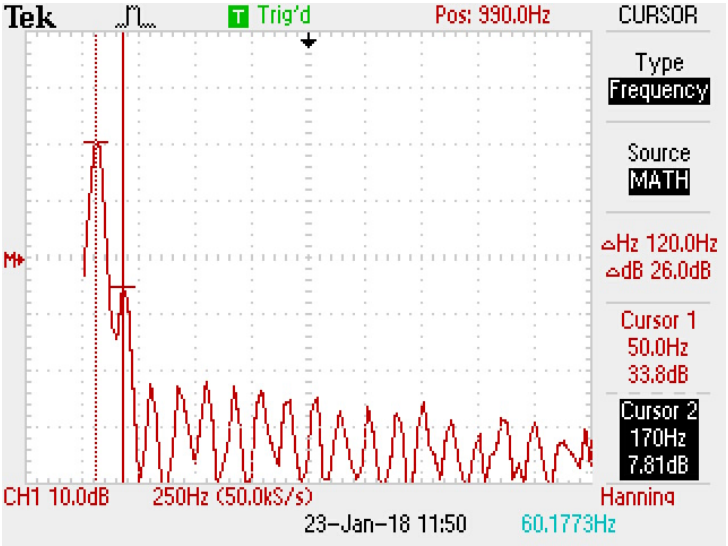


Figure 97: FFT analysis of inverter output with 220Ω load and input voltage of 10V

Table 22 calculates the approximate THD from the 3rd, 5th, 7th, 9th, and 11th harmonics. THD may be as low 5.22% without using feedback to control the output voltage,

which is close to our goal, 4%. As we can see, the 3rd harmonic (180Hz) contributes most of the distortion.

	dB	Gain	Contribution to THD (%)
Fundamental	33.8	48.978	-
3rd	7.91	2.486	94.47%
5th	-10.1	0.313	1.49%
7th	-10.2	0.309	1.46%
9th	-10.7	0.292	1.30%
11th	-10.8	0.288	1.27%
THD	5.22%		

Table 22: THD calculation for 220Ω load and 10V input

Although we can set the exact power supply voltage, the lead acid battery used by our final product will vary based on its charge levels. Therefore, we measured the relationship between output RMS voltage and input DC voltage. The results are shown in Table 23.

Supply Voltage	VCC	Current	Output (RMS)	Efficiency
9.8V	9.0V	7.6A	107V	69.9%
10.8V	9.9V	8.5A	117V	67.8%
12V	11.1V	9.8A	132V	67.3%

Table 23: Output comparison with different input voltages

It is clear from Table 23 that output voltage is positively correlated with input voltage. However, the goal of the project is to make a $115V_{RMS}$ output, and thus neither a $107V_{RMS}$ nor $132V_{RMS}$ output is acceptable. Therefore, a feedback circuit is necessary to control the output, which will be discussed in the following section.

We calculated the efficiency of the inverter circuit by using Equation 13, where η represents the efficiency as a percentage

$$\eta = \frac{P_{out}}{P_{in}} = \frac{V_{out(RMS)}^2 / R_{load}}{V_{in} * I_{in}} * 100\% \tag{13}$$

We found the efficiency was typically around 68% when input voltage was between 10V and 15V. In addition, we found the voltage before the inrush current limiting NTC resistor (voltage from power supply) is significantly different from the voltage after the NTC. There is about a 1V drop over the NTC resistor, which represents a significant power loss (thus, the efficiency after the NTC is typically about 74%). This is also noteworthy because the voltage after the NTC is the actual VCC inverted by the board. However, the NTC is required to prevent the input fuse from blowing when the input capacitors charge, and thus assists the fuse in providing a safety feature for the circuit.

We also found that it takes the inverter some time to turn on and generate a consistent sine wave. The turn on time varies from about 10 seconds up to about 4 minutes. It appears that lower current loads turn on more quickly (for example, at open circuit the inverter starts up immediately), but the time still varies somewhat arbitrarily for any given load. We believe the main cause of this issue is that the step-up transformer

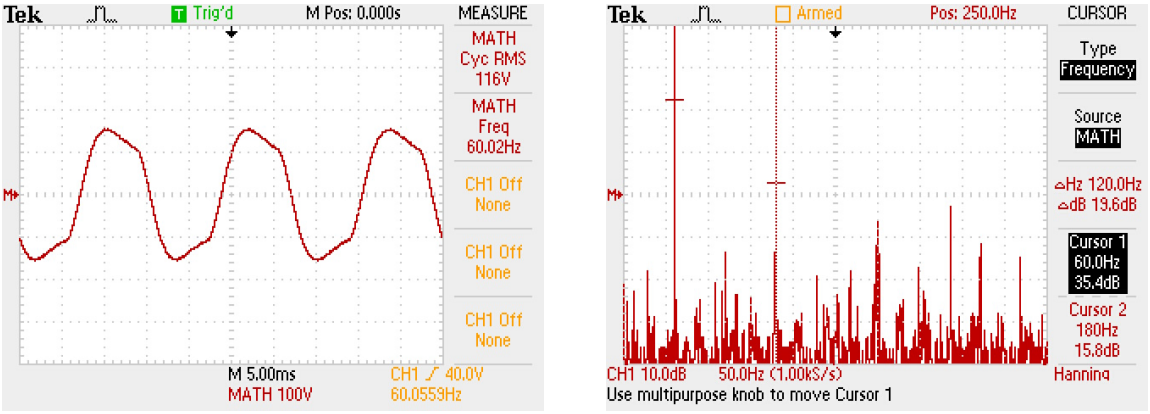
requires several amps (3A or more) of start-up current, in addition to (for example) the 6 to 8A required to power a 60W load. However, the current limit of the bench power supply that we used for testing is only 10A. Thus, during start-up, the power supply hits its current limit, and the supply voltage drops, usually to about 8V but with fluctuations of 1V or so. When it drops below 8V, our half-bridge drivers enter low-voltage drop out and stop outputting to the H-Bridge. At this point, the load current falls to 0A and the supply voltage goes high again, starting the process over. Therefore, the transformer "turns on" and off repeatedly (about once per second) when using the power supply. To solve this problem, the power supply is replaced with a battery, which can provide much more than 10A of current. This will be discussed in a later section.

7.3 Testing 2-Level and 3-Level PWM with Different Resistive Loads

In the last section, a 220Ω power resistor was used as a load, and the circuit performed as expected. Most commercial inverters are meant to power a range of loads with different power requirements, so we also investigated if the inverter circuit could be used with lower power loads. Therefore, we repeated our testing with 300Ω and 400Ω power resistors (respectively). In this section we also discuss the comparison between 2-level PWM and 3-level PWM schemes.

3-level PWM with 300Ω Resistor (44W)

The output when using a 300Ω (44W) resistive load is shown in Figure 98(a). It is clear that the waveform is much more distorted than when using the 220Ω (60W) load resistor. The FFT of the waveform is shown in Figure 98(b).



(a) (b)

Figure 98: (a) 3-level PWM with 300Ω resistor (44W) and (b) Corresponding FFT

Table 24 below calculates the approximate THD, which is 15.36%. However, it is clear from the FFT that there are subharmonics and other frequencies present that are not represented in this calculation, meaning THD is likely significantly larger than calculated here. Regardless, the THD with the 300Ω resistive load is much more significant than THD with the 220Ω resistive load.

	dB	Gain	Contribution to THD (%)
Fundamental	35.4	58.88	-
3rd	15.8	6.17	43.89%
5th	8.4	2.63	7.99%
7th	14.5	5.31	32.54%
9th	9.4	2.95	10.06%
11th	6.8	2.19	5.53%
THD	15.81%		

Table 24: THD calculation for 300Ω load and 10V input

3-level PWM with 400Ω Resistor (33W)

The output distortion becomes visibly worse with a 400Ω (33W) resistor as the load, as shown in Figure 99. In this figure, the green DC voltage is the battery input voltage; the cursor lines are not relevant.

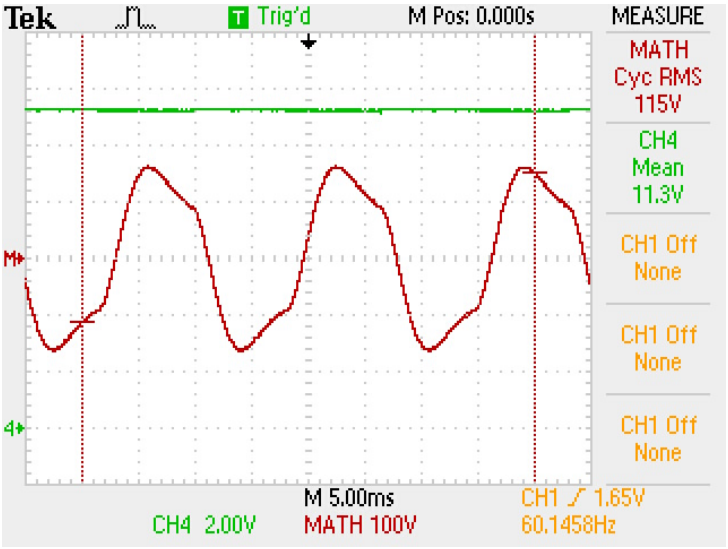


Figure 99: 3-level PWM with 400Ω resistor (33W)

The distortion happens because the load resistor value can affect quality factor Q, which is a dimensionless parameter that describes how underdamped or overdamped an oscillator or resonator is. For an LC low pass filter, the quality factor Q is affected by load resistance, cut-off frequency and capacitance:

$$Q = R_L * w * C \tag{14}$$

When Q is $> (\frac{1}{\sqrt{2}} = 0.707)$, there will be some peaking in the filter response (underdamped system). When Q is < 0.707, the filter response will have a more gentle slope and the roll off will begin sooner (overdamped). Thus, when the load resistor is too big or too small, Q also becomes larger or smaller, and the system can be either overdamped or underdamped, which is shown in Figure 100. An underdamped filter will distort its output by increasing the presence of certain frequencies. An overdamped filter will also distort output, but may also attenuate signals below the cut-off frequency that should

not be attenuated. Therefore, it is very difficult to design a single stage LC filter that can be used for any load.

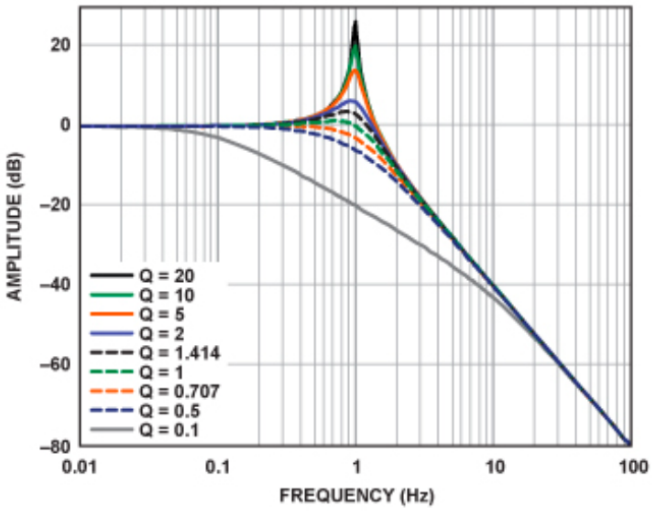


Figure 100: Under/overdamping of LC filter due to improperly matched load resistance

2-Level PWM

After observing that our 3-level PWM switching scheme did not perform as desired with a varying load, we quickly returned to simulation to test the same load changes with 2-level PWM. We found that in simulation, 2-level PWM had a smaller 3rd harmonic for all loads and no significant changes in output even with a varying load. Thus, we updated our microcontroller code for 2-level PWM and tested different load resistors again. Interestingly, for any load resistance, the 2-level PWM inverter output visually resembles a rounded triangle wave, as shown in Figure 101.

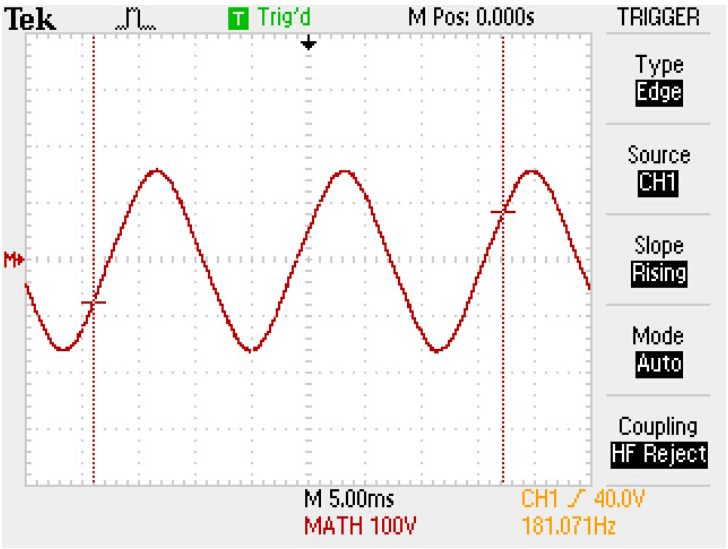


Figure 101: 2-level PWM output measured when load is a 220Ω resistor, input voltage is 11V and feeback is running

The advantage of 2-Level PWM is that the output shape does not noticeably change when with different load resistors, such as 220Ω, 300Ω, and 400Ω. However, although

2-Level PWM performs more consistently when using a purely resistive load, it performs worse when using certain real loads, which will be discussed in the next section.

We also completed a Fourier analysis (FFT) of the inverter output and displayed the signal in the frequency domain to calculate the presence of different total harmonic distortion (THD) for 2-level PWM scheme, shown in Figure 102. Interestingly, the FFT consistently shows that the most prominent harmonic is at 390Hz, which is not an integer multiple of 60Hz.

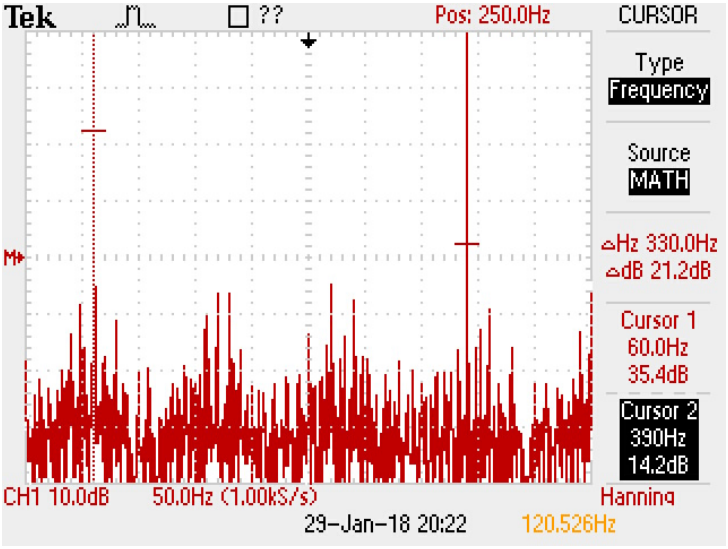


Figure 102: FFT analysis of 2-level PWM inverter output with 220Ω load and input voltage of 10V

Table 25 calculates the approximate THD, which is 13.23%. Thus, THD of 2-level PWM scheme is much smaller than THD of 3-level PWM scheme. However, we didn't choose 2-level PWM scheme because it is not stable when we use a real load, which will be discussed in the next section.

Frequency	dB	Gain	Contribution to THD (%)
Fundamental	35.4	58.88	-
180Hz	12.4	4.17	28.65%
270Hz	12.3	4.12	27.99%
390Hz	14.2	5.13	43.36%
THD	13.23%		

Table 25: THD calculation for 2-level PWM with 220Ω load and 10V input

7.4 Testing Inverter with Household Loads

After testing the circuit with power resistor loads, we tested the circuit with real devices. We chose a 65W laptop as one load, and the parallel combination of two desktop fans (total of 72W) as the other test load. Although we successfully powered the fan and laptop, with both loads the output AC signal is heavily distorted compared to the purely resistive load. Although we expected that these more complicated loads may distort our

output, we did not simulate them, and expected that our 115Hz cut-off frequency filter would be more adaptable. The fan is inductive load, and the laptop adapter is a more complicated load, shown in Figure 103.

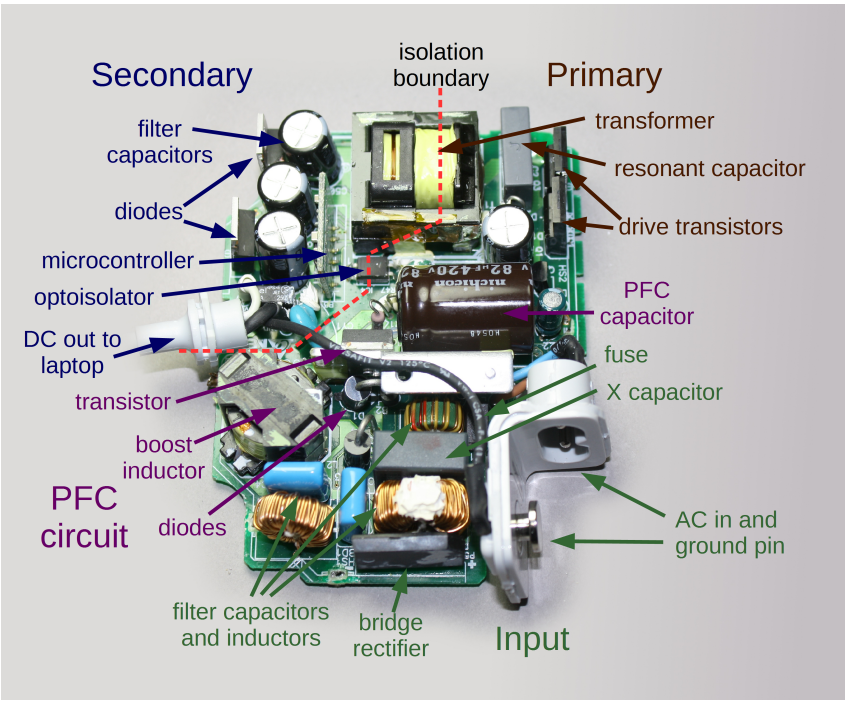
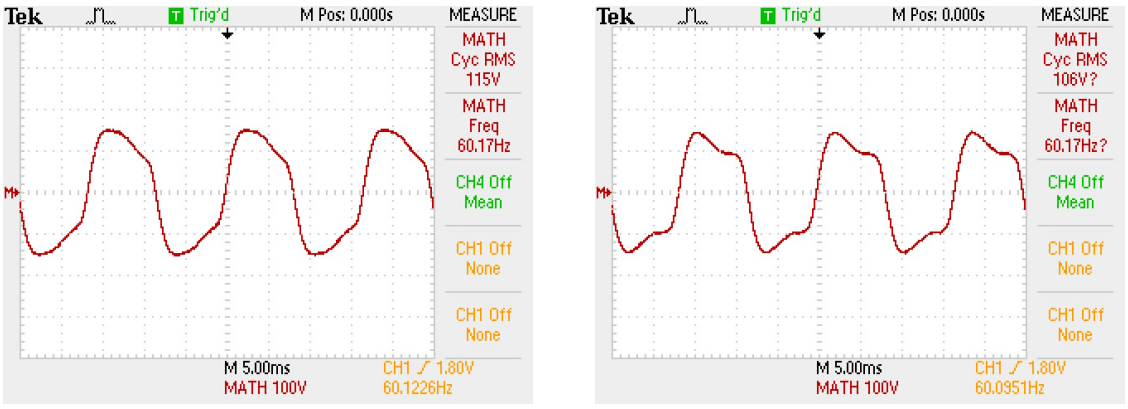


Figure 103: Main components inside the laptop adapter [39]

Powering two fans

The first electronic load we tried was a combination of two desktop fans. We put one 40W fan and one 32W fan in parallel to make an approximately 72W device. Although the inverter powered the two fans and both fans ran constantly, both fans ran more slowly than when connected to mains electricity. The output had significant distortion and was visibly non-sinusoidal. The distortion became larger as the input voltage decreased. Figure 104(a) shows the output signal when the input voltage is 13.5V. Figure 104(b) shows the output signal when input voltage is 10V. It is clear that the 13.5V input test is a closer approximation of a 60Hz sinusoid.

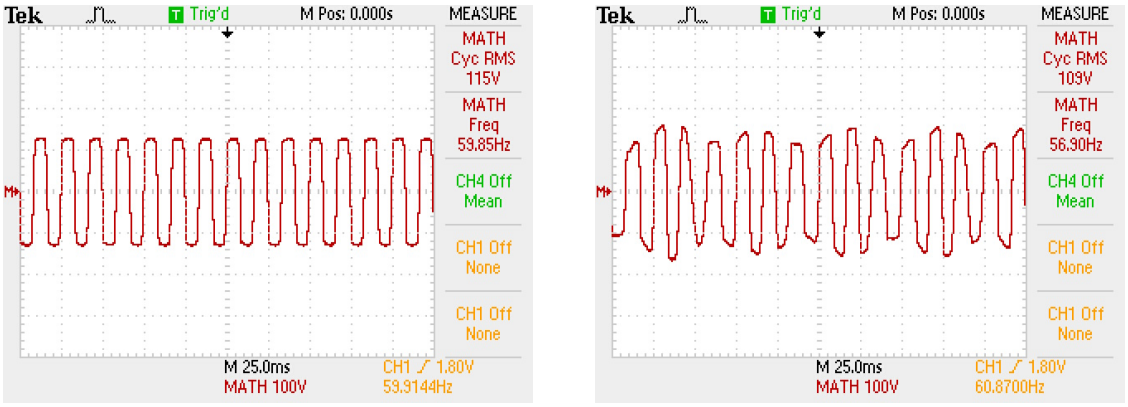


(a) (b)

Figure 104: Output when powering two fans with (a) 13.5V input and (b) 10V input

Powering a laptop

The second real load we tested was a 65W (rated) laptop (via an adapter), and we successfully powered the laptop. The inverter took less than 10 seconds to turn on when the laptop power button was pressed. The laptop did not have a battery, and thus the charger was pulling only enough power to meet the load based on how much the laptop was being used. However, same as the desktop fans, although the laptop could be powered by the inverter PCB, the output wave was distorted, and the distortion was much bigger than that from the fans. As shown in Figure 105(a), the output more closely resembles a rounded square wave than a sine wave. In addition, we also compared the output results when using 2-Level PWM and 3-level PWM. Figure 105(b) is the output when using 2-level PWM.



(a) (b)

Figure 105: Output when powering laptop with (a) 3-level PWM and (b) 2-level PWM

In the last section, we found that 2-level PWM was more consistent than 3-level PWM when using a purely resistive load. However, using the laptop adapter as a load, the voltage regulation system could not hold the output at a stable voltage when using 2-level PWM and thus the output RMS voltage varies by 10V or so. Though this did not have any impact on the functioning of the laptop adapter, instability may have some impact on the functioning of more sensitive devices (for example, a small TV, speaker, etc.). Therefore, we ultimately decided to use 3-level PWM for our inverter.

7.5 Testing Inverter with 12V Battery

All of the previously described tests used a 64V, 10A bench power supply for the DC input because we could set a current limit and control exact input voltage (within about 100mV). After the previously described battery tests, we repeated them with the 12V battery. We connected the battery to the board by two cables and one switch, which is shown in Figure 106.

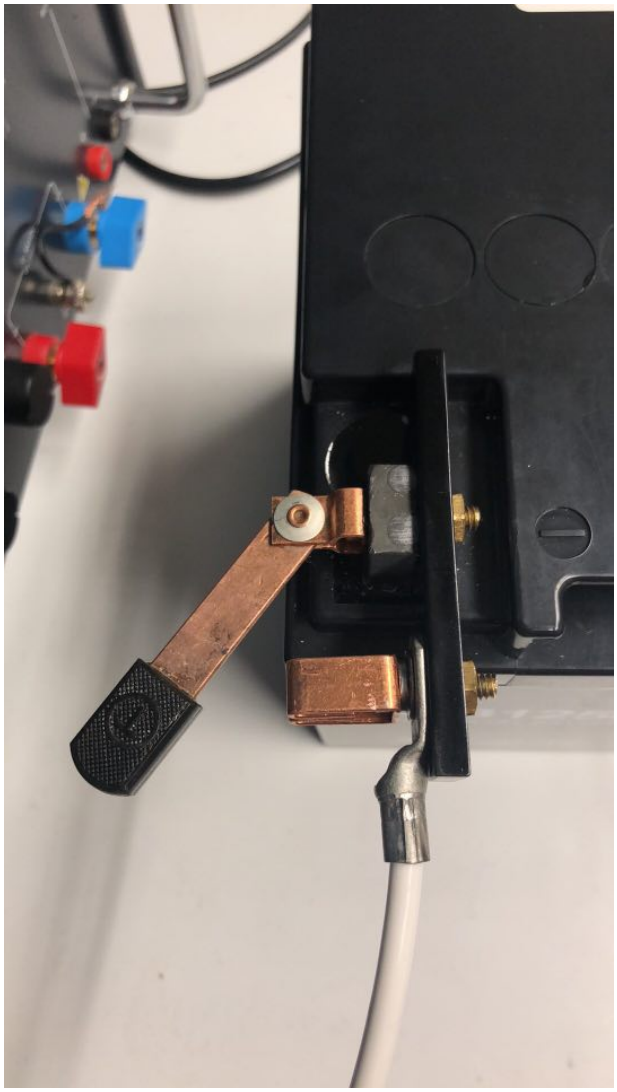


Figure 106: Battery connection switch

When we used the power supply before, the inverter would take some time (sometimes as long as 4 minutes) to turn on because of the 10A current limit. However, when using the battery as the input source, the inverter turns on in about 10 seconds for all loads that we tested.

In conclusion, we successfully powered desktop fans and a laptop with the battery as an input, completing the project goal of powering a 60W AC device with a 12V battery.

7.6 Voltage Amplitude Regulation Testing

Adjusting Output Voltage

The initial test of the amplitude regulation system was to verify that adjusting the PWM scheme would reliably adjust the output voltage of the inverter. Recall from Equation 2 that m_a represents the amplitude modulation ratio:

$$m_a = \frac{V_{sine}}{V_{tri}}$$

The output voltage of the inverter should scale approximately linearly with m_a when m_a is less than or equal to 1, and will continue to increase, though non-linearly, with m_a when it is greater than 1. The relationship between V_{RMS} and m_a for our inverter with a 12V input and 220Ω resistive load is shown in Figure 107. With $m_a > 0.6$ and ≤ 1, V_{RMS} reliably increases by 7V for every increase of 0.1 in m_a. With $m_a \geq 1$, V_{RMS} increases by 9 to 10V for every increase of 0.1 in m_a. However the increase in output voltage drops off steeply above $m_a = 1.2$. At this point, the voltage becomes distinctly more similar to a square wave and V_{RMS} is relatively constant.

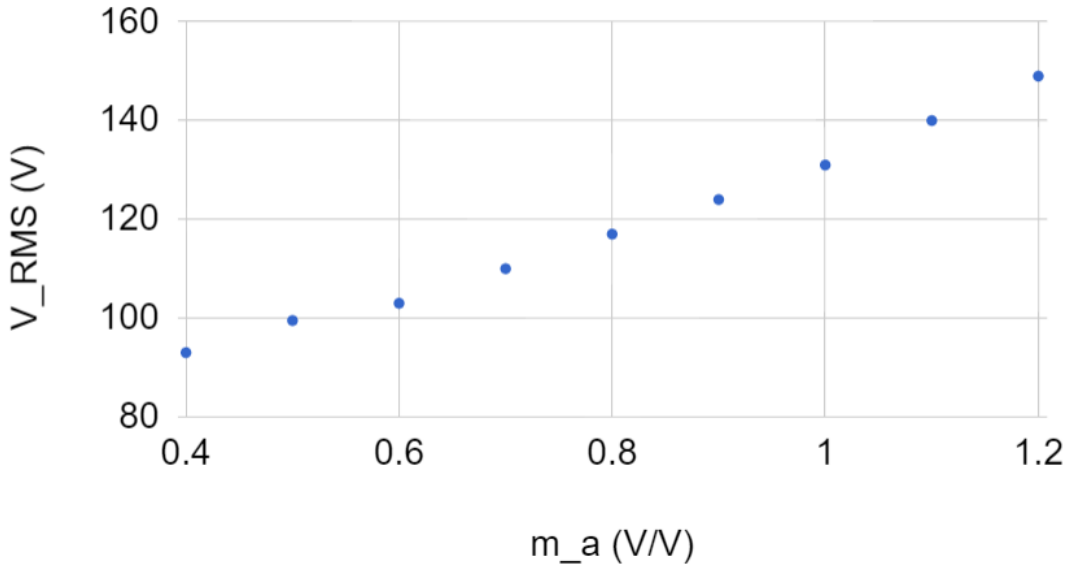


Figure 107: V_{RMS} vs m_a for our inverter with a 12V input and 220Ω resistive load

It was also expected that different values of m_a would result in different harmonic content for the inverter. Given that most harmonic content for our inverter appears at low frequencies, Figure 108 shows the harmonic distortion from the 3rd, 5th, and 7th harmonics vs m_a. Total harmonic distortion (THD) for a waveform with only odd harmonics is calculated as shown in Equation 15 - we used only the fundamental, 3rd, 5th, and 7th harmonic values to calculate the partial harmonic distortion. This graph clearly shows that harmonic content in the waveform increases as m_a decreases. It is noteworthy that there appears to be slightly more distortion with $m_a = 1$. THD is expected to increase as m_a moves away from 1, so in general these results conform to expectations.

$$THD = \sqrt{\frac{\sum_{n=3}^{n_{max}} {V_n}^2}{{V_1}^2}} \tag{15}$$

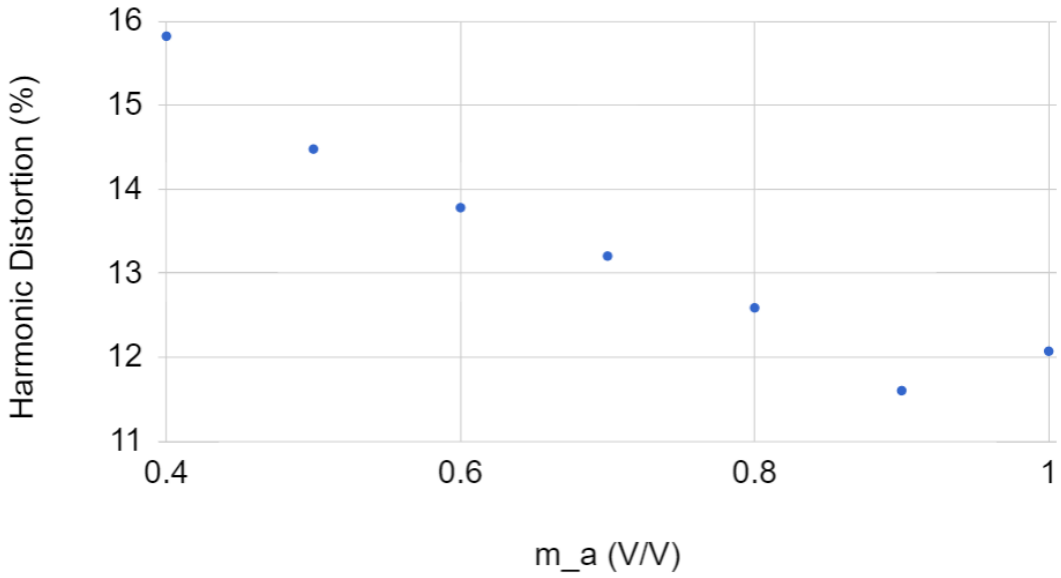
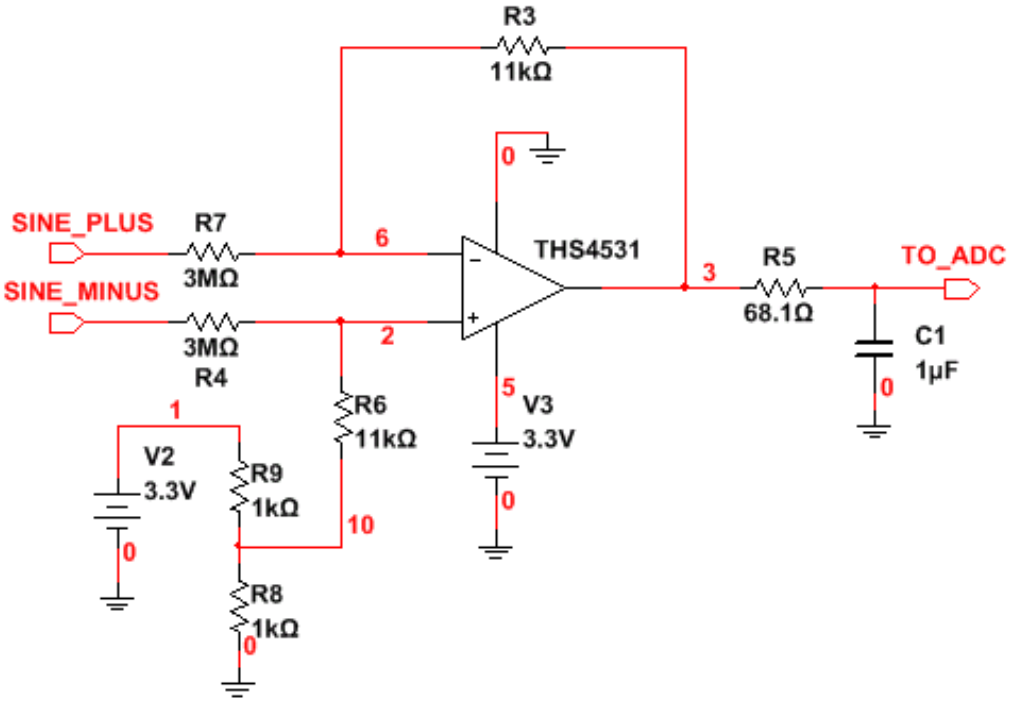


Figure 108: Harmonic distortion of output voltage from the 3rd, 5th, and 7th harmonics vs m_a for our inverter with a 12V input and 220Ω resistive load

Feedback Circuit Functionality

The amplitude regulation system is based off of the differential amplifier feedback circuit shown in Figure 49 in Section 4, shown again here.



Initial tests of this circuit were problematic - significant noise (in the form of a 1MHz sinusoid) was seen to be coupled on the feedback input to the ADC, as seen in Figure 109. A closer look at the noise on the feedback signal is shown in Figure 110.

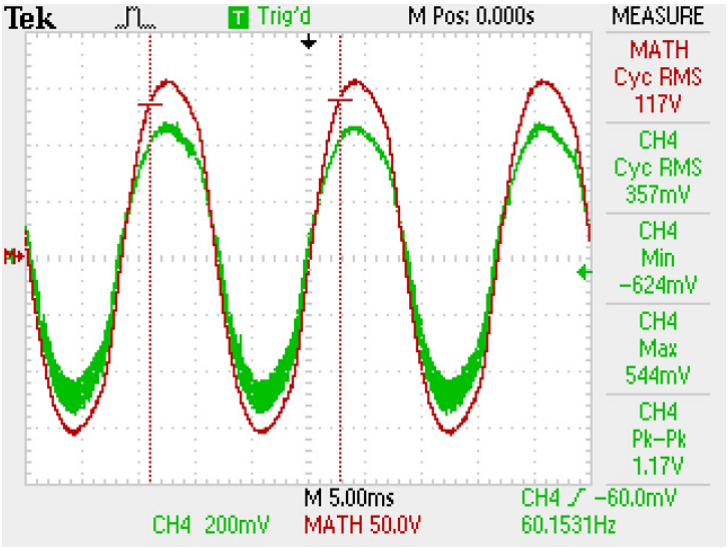


Figure 109: $117V_{RMS}$ inverter output (red, clean signal) and feedback signal to ADC with 1MHz noise coupled (green, noisy signal)

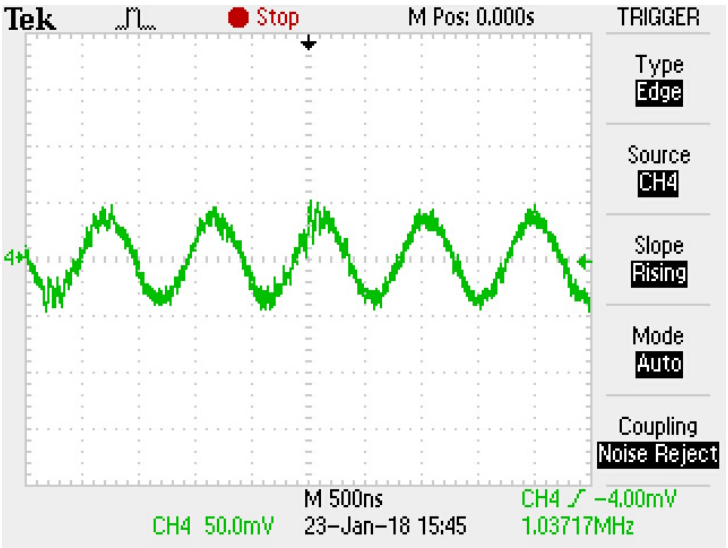


Figure 110: 1MHz noise coupled to ADC input

It was initially believed that the (approximately) 1MHz noise was a function of a clock signal from the microcontroller on the Launchpad being coupled from the ADC onto the PCB. After some experimenting, however, it was determined that the signal was being generated by oscillations between the output of the op-amp and the lowpass filter on the amplifier output. The exact cause of the noise was not determined, but it was assumed the RC circuit had become an oscillator of sorts due to unfortunate choices of filter values. Because the filter was chosen with a relatively high cut-off frequency of 530Hz, the initial solution was to increase the resistor value and thus lower the cut-off frequency. Thus, the 3kΩ resistor was replaced with a 10kΩ resistor, lowering the cut-off frequency to 160Hz. This reduced the 1MHz noise somewhat, but did not eliminate it. The next effort was to remove the capacitor, thus eliminating the filter entirely. This was shown to entirely eliminate the 1MHz noise without introducing any additional noise - in other words, the filter was not only detrimental, but unnecessary, because with a standard resistive load,

the feedback signal showed no other noise (although some switching frequency transients do appear to be coupled from the nearby H-Bridge), as shown in Figure 111.

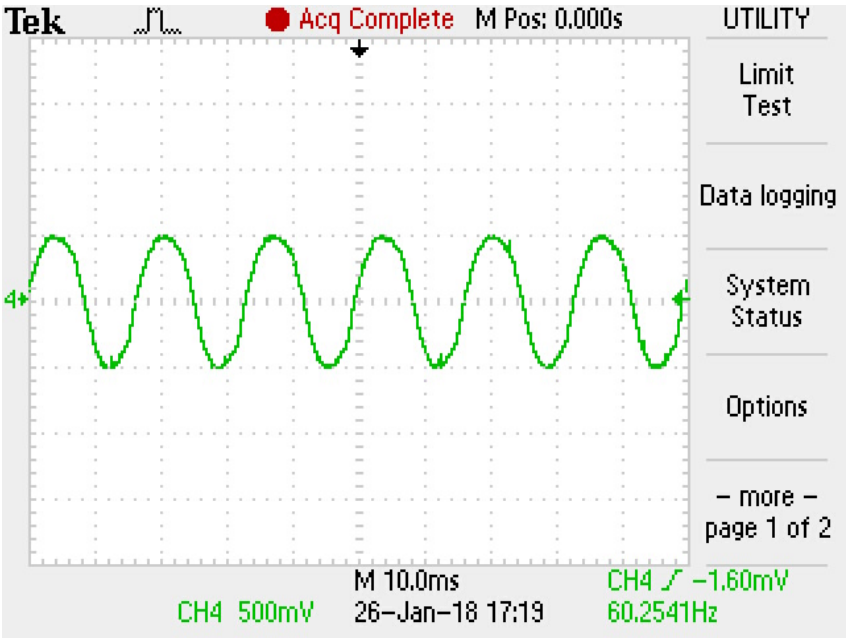


Figure 111: Feedback signal without RC lowpass filter

Calculating Output Voltage Amplitude

The next phase in the feedback system is the ADC. Figure 112 shows the sine wave sensed by the ADC, which is viewed through the C2000 debugger in TI Code Composer Studio, and measured in ADC codes. On the C2000's 12-bit ADC, each ADC code corresponds to $3.3\text{V}/(2^{12}) = 0.806\text{mV}$. With the feedback circuit having a gain of approximately $11\text{k}\Omega/3\text{M}\Omega = 0.00367$ between the inverter output and ADC input, each ADC code represents approximately $0.806\text{mV}/0.00367 = 220\text{mV}$/code on the inverter output. The waveform is visibly quite similar to the actual output of the inverter, with its characteristic third-harmonic peaks.

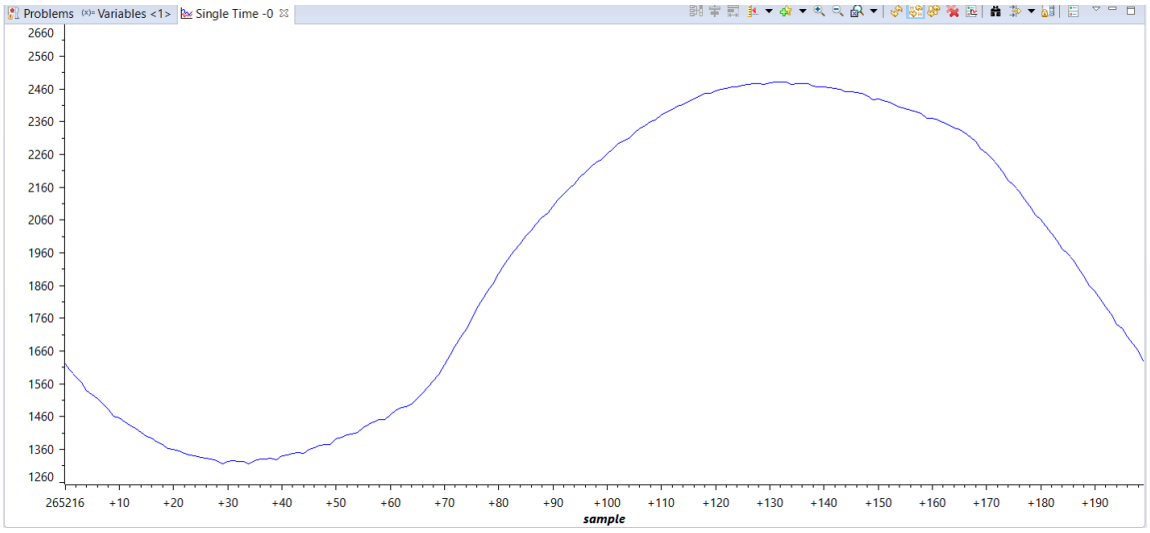


Figure 112: ADC reading, viewed through debugger in CCS

The inverter was originally specified to maintain a voltage of 115V±10%, i.e. 103.5V to 126.5V. Thus, it is expected that for a constant output voltage, the microcontroller should sense an amplitude with ripple of well less than 10% of 115V, i.e. 11.5V. 11.5V corresponds to about 52 ADC codes, so the ripple on the calculated amplitude should be much less than this. The initial assumption was that it would be preferable for the ripple to be less than 5 or 10 codes, but this would have to be verified by determining what maximum level of ripple could occur without being reflected on the inverter output.

Shown in Figure 113 is the IQ amplitude calculation with a 10-sample moving average. At 25kSps, a 10-sample average contains 0.4ms of data, less than one fortieth of a 60Hz sine wave cycle. The 10-sample moving average method of lowpass filtering proved adequate (with a swing of less than 5 codes) for initial tests of the amplitude calculation system when a function generator based sine wave was fed into the ADC. However, because of the non-negligible third harmonic component in our inverter's output waveform, the IQ sampling amplitude calculation method appeared to be somewhat volatile. For our circuit, over the course of two 60Hz cycles, this reading has a swing of nearly 100 ADC codes, or about 100 codes * (220mV/code) = 22V, which is too large to be acceptable.

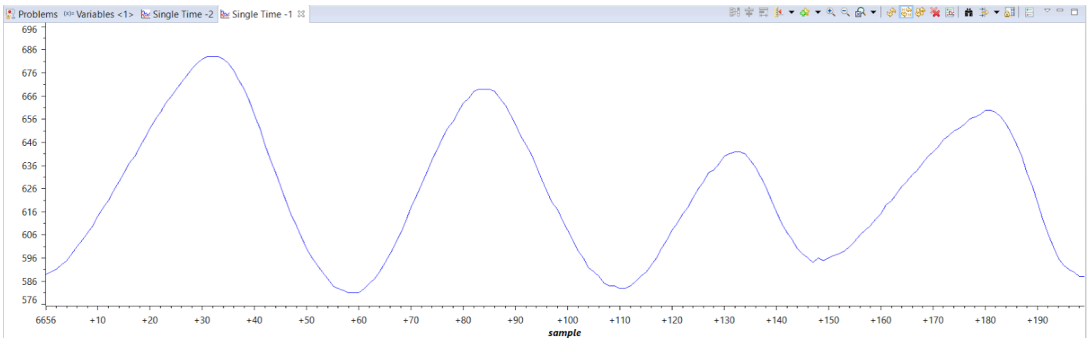


Figure 113: Amplitude calculations with 10 sample rolling average (swing of 100 ADC codes)

To increase accuracy, a 500-sample moving average was used. At 25kSps, this represents 20ms of data, which is greater than the length of a 60Hz cycle (16.66ms). This amplitude calculation, shown in Figure 114, had a ripple of less than 5 codes, and was thus considered adequate to begin testing the feedback system with. It is worth noting that a potential consequence of using a larger moving average is that it will slow down the impact that the feedback system has on the output of the inverter. For example, if the calculated amplitude is less than the setpoint, the PID loop will increase gain and attempt to raise the output voltage to the setpoint. However, even if the output voltage immediately rises, only a few of the samples in the 500 sample moving average buffer will reflect this, and the amplitude calculated by the program will not reflect the true output of the inverter. This may cause the PID gain to further increase, as it senses that the gain it is applying is not strong enough to reach the voltage setpoint. This could cause overshoot in the output voltage until the buffered average finally reflects the true amplitude. However, with a correctly tuned PID, these issues can likely be avoided, though potentially at the expense of the overall response time of the PID loop.

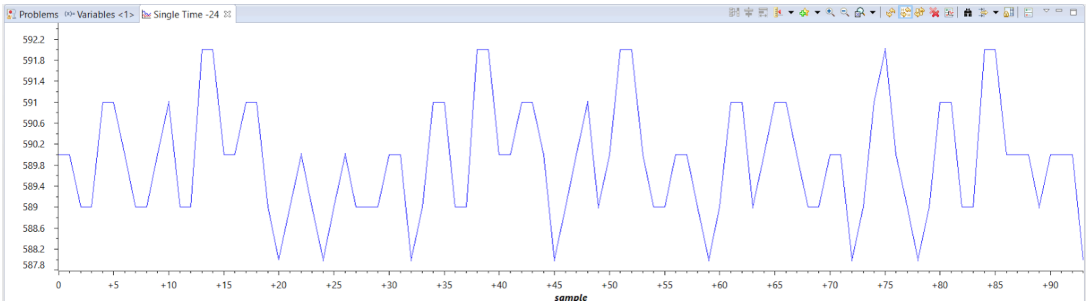


Figure 114: Amplitude calculations with 500 sample rolling average (swing of 5 ADC codes)

Functionality of Full System

After successfully calculating the output voltage amplitude, the software PID controller was instantiated to adjust m_a based on the difference between the measured output voltage and a voltage setpoint. With a new gain calculated after every ADC sample (25kSps), $K_P = 0.01$, $K_i = 0.03$, and $K_d = 0$, the system maintains output voltage between $114V_{RMS}$ and $116V_{RMS}$ for any steady state input voltage between 9.5V and 14V. The chosen PID constants give a relatively slow response - a step-like change in input voltage (for example, from 14V to 10V in less than a second) may cause the input voltage to go as high as 120V or as low as 110V, with a steady state of 115V reached within 2 to 3 seconds. This is acceptable behavior, however, because the input will never see a step change in voltage from the battery. Rather, it will see a steady decline over the course of minutes and hours, and the feedback system is designed with this behavior in mind. Figure 115 shows the output signal with the feedback circuit, which is exactly $115V_{rms}$. It was found that with low input voltages (close to 10V), THD is close to 5% as previously measured. However, as input voltage increases (and m_a decreases in response), THD was found to increase up to 17% (with input voltages close to 15V).

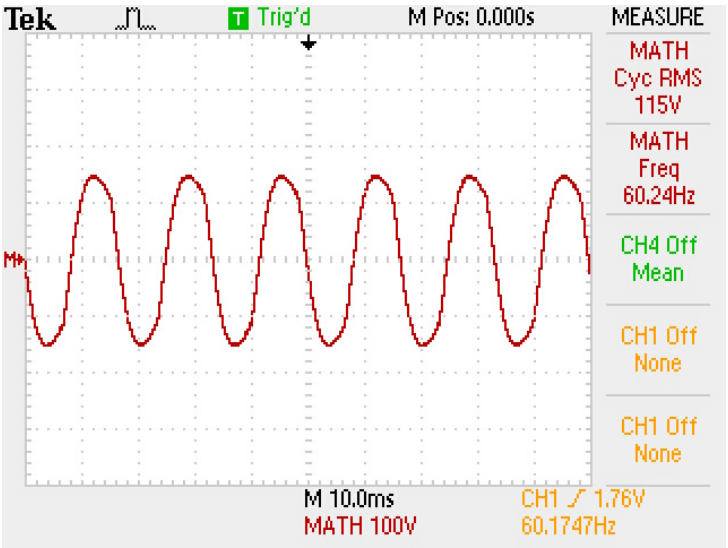


Figure 115: Voltage measured across the power resistor with feedback circuit

7.7 Reliability Calculations

The factors that highlight the importance of reliability prediction calculations for our inverter are stated below:

1. The reliability prediction highlights the failure rates of the components in a design and can be used as a guide to improve the highest contributors to failure of the system.

2. Over-stressed parts of the system can be determined, allowing for improvement to ensure longevity of the system.

3. The need for back-up systems can be predicted by the reliability prediction calculation which impacts manufacturing costs and influences the market price of the product.

Our reference for calculating the reliability prediction is the MILITARY HANDBOOK RELIABILITY PREDICTION OF ELECTRONIC EQUIPMENT MIL-HDBK-217F (2 DECEMBER 1991) [40]. There are two major methods of calculating the reliability prediction. The first is the Part Stress Analysis method, which uses detailed information about the operating condition of each component to estimate its failure rate. The second method is the Part Counts Method, which we have selected, which does not require information about the operating conditions of the circuit, and instead uses standard reliability data for different classes of devices. The Part Counts method is a good first-cut calculation that does not require the significant time investment of the Part Stress Analysis method. The primary equation we will be using is shown below:

$$\lambda_{\text{EQUIP}} = \sum_{i=1}^{i=n} N_i(\lambda_g \pi_Q)_i \tag{16}$$

λ_{EQUIP} = Total equipment failure rate (Failures/10^6 hours)
λ_g = Generic Failure rate for the ith generic part (Failures/10^6 hours)
π_Q = Quality factor for the i^{th} generic part
N_i = Quantity of i^{th} generic part
n = Number of different generic part categories in the equipment

There are two conditions that have to be determined when using this equation. The first condition is the environment in which the inverter will operate in. We have chosen the environment to be "Ground, Benign" (G_B). The next condition is to determine the level of testing that the components used in our inverter have undergone. We have chosen the "Lower" testing condition because most of our components were purchased from Digikey and did not undergo military-level testing screenings. With these conditions set, we calculated the failure rates for each of our components and calculated the overall failure rate of our inverter. All the values are rated at 50°C. The results have been highlighted in Table 26. This table also includes the operating power dissipation of each major component, because although the power dissipation was not used in our reliability calculations, it would have an impact on reliability in the field. The gate driver, microcontroller and the SMPS were all TI products and their failure rates were taken from TI's online reliability calculator [41].

Part	Quantity	Power Dissipation	Generic Failure Rate λ_g	Quality Factor π_Q	Failure Rate of Component Part λ_{EQUIP} [Equation 16]
MOSFET	4	0.028W	0.014	8	0.448
Driver	2	-	-	-	0.000862
Bootstrap Capacitor	2	-	0.0017	10	0.034
Terminal Block (2x8) gauge	4	-	0.062	3.4	0.8432
1MΩ Feedbacck resistor	6	0.125W	0.0037	10	0.222
11kΩ Feedback resistor	2	0.125W	0.0037	10	0.074
12V to 5V SMPS	1	-	-	-	0.000212
NTC Thermistor	1	5.7W	0.0014	10	0.014
Transformer	1	21.875W	0.053	3	0.159
Microcontroller	1	-	-	-	0.0023
Input Capacitance	5	0.052	0.0013	10	0.065
Output Filter Capacitor 4uF	1	-	0.0007	10	0.007
Output Filter Capacitor 0.3uF	1	-	0.0007	10	0.007
Output Filter Inductor 320mH	1	2.125W	0.000032	3	0.000096
Total Equpment Failure Rate					1.8766 Failures/10^6 hours
MTBF					532869 hours

Table 26: Reliability calculation of components

The power dissipation data in this table shows that the transformer, which dissipates 21.875W, is by far the greatest contributor to the inefficiency of the inverter. The NTC thermistor also adds a major contribution at 5.7W, as does the output filter inductor at 2.125W.

The Total Equipment Failure Rate is calculated by summing the failure rates of all the components calculated in Table 26 above. As a result, the Total Equipment Failure Rate is 1.8766 Failures/10^6 hrs. The Mean Time Between Failure (MTBF) is calculated to be 532869 hours which is approximately 60 years.

Our inverter is expected to run for a maximum of 8 hours a day with a rated reliability of 92%. Given the reliability formula below, we then calculated the life expectancy of our inverter. Our inverter is to have a life expectancy of 15 years.

$$R(t) = e^{-t/MTBF}$$

where t = Operation of inverter (yr) × Life expectancy of inverter
e = 2.718
MTBF = 532869 hrs

7.8 Conclusions

In summary, the inverter prototype was fully implemented on a PCB and was found to be functional for all loads with which it was tested. The table below summarizes how well the design meets our product specifications.

Initial Requirement	Result for Prototype
Pure sine wave output, with THD < 4 %	This specification was not fully satisfied. With 3-level PWM, THD is $\geq 5\%$ for all loads
Efficiency of 80-90% at nominal power	This specification was not fully satisfied. Maximum recorded efficiency is 69.9%, and is typically between 67 and 68%
Nominal power output of at least 60W	This specification was fully satisfied. The inverter functions best with a load that requires close to 60W
Output voltage 115VAC$\pm$10%	This specification was fully satisfied. The inverter outputs between $114V_{RMS}$ and $116V_{RMS}$ over its full input range, regardless of the type of load
Output frequency 60Hz$\pm$0.1%	This specification appears to be met satisfactorily. When measured with an oscilloscope, the output of the inverter is near 60Hz, but an exact value is not given
Output current at least 0.5A, nominal	This specification was fully satisfied. The inverter is capable of sourcing more than 0.5A

Table 27: Product specifications

8 Conclusions and Recommendations

8.1 Overall Summary and Achievements

This MQP resulted in the successful design, simulation, construction, and testing of an off-grid power inverter. The inverter was built with a custom PCB, with microcontroller-based 3-level PWM controlling an H-Bridge switching circuit, and voltage stepped up to mains levels through a 60Hz step-up transformer. The design also successfully implemented a feedback-based voltage amplitude regulation system, which was recommended by previous MQP groups that built inverters. The inverter can reliably power a small electronic load such as a laptop or desktop fan.

Many of the major components and design choices for this MQP were successful - the chosen H-Bridge circuit and corresponding microcontroller, MOSFETs, and MOSFET drivers performed as expected. Additionally, the voltage amplitude control system was successful at maintaining a stable output voltage for all loads. That said, some design choices proved to be less suitable, such as the choice of the low frequency topology, which lowered overall efficiency and caused start-up delays due to the large magnetizing current required for the 60Hz transformer. Additionally, the THD of our final output waveform was significantly higher than desired, at 5.22% (under ideal conditions with a purely resistive load) and often much greater. Similarly, the efficiency of the inverter was lower than desired, at about 70% or less. The inefficiency was due primarily to the low frequency transformer. Despite this, the inverter is capable of sourcing its nominal power and successfully powering the loads it was intended to power.

8.2 Lessons Learned

Our team learned several key lessons throughout the course of completing this MQP. Possibly most importantly was that the drawbacks to the low frequency inverter topology should not be understated. The transformer required for our design was very large and would require our 60W inverter to be larger than common commercial inverters that supply hundreds of watts. Additionally, the transformer proved to be a major source of wasted energy with a total power dissipation of 21.875W. This may be due in part to the use of a step-down transformer as a step-up.

This relates to another lesson learned for a design problem such as ours (i.e. one to which partial or complete solutions exist, and which does not have unusual design constraints). This lesson is that, when possible, design choices should take advantage of relatively standard components and standard design options. Because of our low frequency topology, we were required to use expensive H-Bridge MOSFETs to take advantage of their low on resistance - this part of the circuit performed well but would increase the cost of a commercial product. The low frequency topology also required us to use an abnormally heavy, expensive, and inefficient step-down transformer in reverse. This led to other problems with our design, and indicates that the more commonly used high-frequency transformer topology would likely have been a better design decision. That said, there will be times when a design, especially a groundbreaking design or one with unusual constraints, will require uncommon or custom-designed parts.

We learned in this project that it is crucial to carefully inspect the datasheets of all parts, and especially to understand the interfaces between parts. For example, we had a mishap when we ordered the wrong MOSFET driver, believing that it was compatible with our microcontroller because both advertised “CMOS logic levels”. However, we

found later that the driver actually used 10V CMOS, which is not compatible with the 3.3V CMOS of the microcontroller.

Another lesson learned was to understand the limitations of circuit simulation. Our team spent close to two weeks attempting to categorize and understand the cause of H-Bridge shoot-through present in our simulations in Multisim, to no avail. While we still do not understand the exact cause, the simulated shoot through is likely an artifact of either an imperfect MOSFET Spice model, or of fast switching with discrete time simulations. Regardless, our understanding of the circuit led us to believe that despite this simulation result, the physical prototype would work as expected, which it ultimately did.

One other lesson specific to a power electronics project is to more thoroughly characterize the properties of the circuit's load, preferably early in the project. With the 60W resistive dummy load, our circuit worked as well as it was simulated to. However, the output of our inverter was found to be imperfect with non-resistive or lower powered loads (i.e. more 3rd harmonic content with inductive fan loads, and square wave output with AC/DC adapter load). We may have been able to correct for this earlier in the project if we had been able to simulate our inverter with similar loads.

In this project, the team learned several engineering skills that are useful to any project. For example, temperature calculations for different components on our PCB were crucial for proper component selection and a useful exercise in general. Additionally, the reliability calculations we completed will likely be required for almost any commercial product design we may eventually be involved in.

8.3 Recommendations

An interesting extension of this project for future MQPs would be to investigate more sophisticated feedback control for the sake of actively filtering the output voltage and current. This feedback system would be much more complicated than the one used by our circuit, but would greatly improve the ability of the inverter to supply a stable voltage with low THD, and allow all loads to run at high efficiency. A sophisticated feedback system may also help to eliminate the need for a large output filter, lowering the cost of the circuit.

We recommend that any similar inverter related MQP not consider the Low Frequency topology. Instead, the High Frequency Transformer or Transformerless topologies should be used. These topologies should contribute to higher efficiency and lower component cost. It is possible to (as done by previous MQPs) use the DC/DC conversion stage from an existing inverter, or build a DC/DC converter if the team believes it is not an unreasonable amount of work to do so and can be completed within the allotted MQP time.

A project building off of our work should also reconsider the choice of input fuse and NTC inrush current limiter that we used. The NTC is required to protect our 15A fuse from blowing when starting the circuit. However, the NTC is a major source of inefficiency. One possible solution is to remove the input fuse entirely (or choose a larger fuse) and to remove the NTC. Either of these solutions may result in a circuit that is less safe, so additional considerations must be taken. Another possible solution is to design a small circuit that allows the circuit to start up with a small resistance in series with the fuse to prevent the fuse from blowing, then bypasses the resistor (through a switch) once the circuit is operating fully.

If a future inverter or other power electronics project team is interested in building a very high efficiency converter, it would be interesting to explore more precise switching methods. For example, a team could attempt to maximize efficiency with deadtime control, because lowering deadtime decreases MOSFET reverse conduction losses [42].

Future inverter/power electronics related MQPs could learn quite a bit about real commercial product design by adding various safety features to their device. Commercial inverters have a variety of safety features listed in Table 4 in Section 2, which would be interesting to implement. Additionally, our inverter specifically would benefit from overvoltage protection on the input because our input capacitor bank is only rated for 16V. Thus, a new safety feature could meet this need by immediately disconnecting any output exceeding 15V, and possibly discharging the capacitors to 15V or less.

The applications of inverters to electric vehicles would also be an interesting topic of exploration for a future MQP. This may give a similar but not identical set of design constraints and has not been explored by any previous MQP.

Future MQPs could explore new, groundbreaking technologies and their use in Power Electronics projects. One possible suggestion is the exploration of GaNFETs and their possible applications in commercial products.

References

[1] "Projected number of households using off-grid solar worldwide from 2010 to 2020", Tech. Rep., 2016. [Online]. Available: https://www.statista.com/statistics/535180/forecast-for-offgrid-solar-household-users-worldwide/.

[2] System examples. [Online]. Available: http://www.backwoodssolar.com/learning-center/systems-portfolio/off-grid-system-examples.

[3] Off grid solar block diagram. [Online]. Available: https://mozaw.com/wp-content/uploads/2017/02/off-grid-solar-system-1024x557.jpg.

[4] I. Crowley and H. F. Leung, "Pwm techniques: A pure sine wave inverter", Tech. Rep., 2011. [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-042711-190851/unrestricted/PWM_Techniques_final.pdf.

[5] O. Rich and W. Chapman, "Three-level pwm dc/ac inverter using a microcontroller", 2012. [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-042412-155252/unrestricted/MQP_Report_Final_2012-04-26.pdf.

[6] Grid tied pv systems. [Online]. Available: http://energyinformative.org/wp-content/uploads/2012/05/grid-tied-solar-system.png.

[7] Inverter output waves. [Online]. Available: http://www.reuk.co.uk//OtherImages/square-modified-sinewave.jpg.

[8] Choosing an inverter for your solar power system. [Online]. Available: http://www.solar-facts.com/inverters/inverter-choice.php.

[9] D. W. Hart, Power electronics, International ed. New York, NY: McGraw-Hill, 2011.

[10] A. Beaudet, Pure sine wave vs. modified sine wave inverters – what's the difference?, 2015. [Online]. Available: https://www.altestore.com/blog/2015/10/pure-sine-wave-vs-modified-sine-wave-whats-the-difference/.

[11] Inverter basics and selecting the right model. [Online]. Available: https://www.solar-electric.com/learning-center/inverters/inverter-basics-selection.html.

[12] Single phase solar inverters. [Online]. Available: https://www.solaredge.com/us/products/pv-inverter/single-phase#/.

[13] A. Singh and J. VS, "Voltage fed full bridge dc-dc and dc-ac converter for high-frequency inverter using c2000", Tech. Rep., 2014. [Online]. Available: http://www.ti.com/lit/an/sprabw0b/sprabw0b.pdf.

[14] H. Robertson, Optimizing photovoltaic systems, 2010. [Online]. Available: https://www.electronicproducts.com/Power_Products/Invertors/Optimizing_photovoltaic_systems.aspx.

[15] Elite 2000 pro. [Online]. Available: http://wagan.com/wagan-tech/power-inverters/elite-200w-pro.html.

[16] B. Burger and D. Kranzer, "Extreme high efficiency pv-power converters", English, 2009, pp. 1–13. [Online]. Available: http://ieeexplore.ieee.org/document/5279115.

[17] "Pwm with constant duty cycle", [Online]. Available: https://developer.android.com/things/images/pwm-duty.png.

[18] Squarewave fourier components. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/e/ec/Squarewave01CJC.png.

[19] H-bridge switch positions. [Online]. Available: https://i.stack.imgur.com/s9Uvc.png.

[20] N. Mohan, Power electronics : converters, applications, and design, English. United States: Knovel, 2003, ISBN: 9780471429081. [Online]. Available: http://catalog.hathitrust.org/Record/006834951.

[21] F. L. Luo and H. Ye, Advanced DC/AC Inverters, English, 1st ed. Baton Rouge: CRC Press, 2017, vol. 1, ISBN: 9781138072848. [Online]. Available: http://www.crcnetbase.com/isbn/9781466511385.

[22] Switching losses. [Online]. Available: https://www.maximintegrated.com/en/images/appnotes/4266/4266Fig04.gif.

[23] H-bridges - the basics. [Online]. Available: http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/.

[24] How to read a fet datasheet. [Online]. Available: http://www.physics.unlv.edu/~bill/PHYS483/fet_datasheet.pdf.

[25] M. H. Rashid, Power electronics handbook, 3. ed. Amsterdam [u.a.]: Elsevier, 2011, ISBN: 0123820367.

[26] Insulated gate bipolar transistor. [Online]. Available: http://www.electronics-tutorials.ws/power/insulated-gate-bipolar-transistor.html.

[27] E. Persson, A. Bricconi, and F. Grawert, "Gan in a silicon world: Competition or coexistence?", Tech. Rep., 2016. [Online]. Available: https://www.infineon.com/dgdl/Infineon-Presentation_GaN_GalliumNitride_APEC2016-AP-v01_00-EN.pdf?fileId=5546d46253a864fe0153d0a8f85132c5.

[28] J. Strydom, The egan fet-silicon power shoot-out vol. 7: Buck converters, 2012. [Online]. Available: http://www.powerelectronics.com/gan-transistors/egan-fet-silicon-power-shoot-out-vol-7-buck-converters.

[29] T. Wescott, Pid without a phd, English, 2000. [Online]. Available: https://search.proquest.com/docview/218514962.

[30] M. Gonzalez, V. Cardenas, and F. Pazos, "Dq transformation development for single-phase systems to compensate harmonic distortion and reactive power", English, IEEE, 2004, pp. 177–182. DOI: 10.1109/CIEP.2004.1437575. [Online]. Available: http://ieeexplore.ieee.org/document/1437575.

[31] M. Kuisma, I/q data for dummies, 2017. [Online]. Available: http://whiteboard.ping.se/SDR/IQ.

[32] Ul 1741. [Online]. Available: https://standardscatalog.ul.com/standards/en/standard_1741_2.

[33] Ul 458, 2015. [Online]. Available: https://standardscatalog.ul.com/standards/en/standard_458_6.

[34] Tms320x2802x, 2803x piccolo enhanced pulse width modulator (epwm) module, 2011. [Online]. Available: http://www.ti.com/lit/ug/spruge9e/spruge9e.pdf.

[35] Tms320f802x piccolo microcontrollers, 2017. [Online]. Available: http://www.ti.com/lit/ds/symlink/tms320f28027.pdf.

[36] Isolation bootstrap calculator. [Online]. Available: https://www.silabs.com/tools/Pages/bootstrap-calculator.aspx.

[37] Ultracapacitor amp; supercapacitor frequently asked questions. [Online]. Available: https://www.tecategroup.com/ultracapacitors-supercapacitors/ultracapacitor-FAQ.php.

[38] Ipc - 2152. [Online]. Available: http://electronica.ugr.es/~amroldan/cursos/2014/pcb/modulos/temas/IPC2152.pdf.

[39] The components inside an apple macbook 85w power supply, 2015. [Online]. Available: http://www.righto.com/2015/11/macbook-charger-teardown-surprising.html.

[40] D. of Defence USA, Military Handbook Reliability Prediction of Electronic Equipment, Notice 2. Department of Defence United States of America, 1991.

[41] Ti reliability calculator. [Online]. Available: http://www.ti.com/quality/docs/estimator.tsp?OPN=UCC27201DR&CPN=&partNumber=UCC27201#resultstable.

[42] J. Strydom and D. Reusch, "Dead-time optimization for maximum efficiency", Tech. Rep., 2013. [Online]. Available: https://epc-co.com/epc/Portals/0/epc/documents/papers/Dead-Time%20Optimization%20for%20Maximum%20Efficiency.pdf.

[43] Types of solar panels. [Online]. Available: https://www.ecokarma.net/solar/types-of-solar-panels/.

[44] A. Alvarez, 3 types of solar panels: Pros and cons, 2013. [Online]. Available: https://www.angieslist.com/articles/3-types-solar-panels-pros-and-cons.htm.

[45] T. Agarwal, Solar charge controller types, functions, and applications. [Online]. Available: https://www.angieslist.com/articles/3-types-solar-panels-pros-and-cons.htm.

[46] S. Davis, Solar system efficiency: Maximum power point tracking is key, 2015. [Online]. Available: http://www.powerelectronics.com/solar/solar-system-efficiency-maximum-power-point-tracking-key.

[47] Battery basics: A layman's guide to batteries, 2017. [Online]. Available: https://www.batterystuff.com/kb/articles/battery-articles/battery-basics.html.

[48] Introduction to lead acid batteries. [Online]. Available: http://www.batterytender.com/intro-to-lead-acid-batteries.

[49] Inverter and battery technology, 2013. [Online]. Available: http://powersmartsolar.co.nz/blog/id/442.

[50] L. Lalonde, Don't judge a solar pv system's efficacy by inverter efficiency alone, 2011. [Online]. Available: http://www.electronicdesign.com/energy/don-t-judge-solar-pv-system-s-efficacy-inverter-efficiency-alone.

[51] String inverters vs. microinverters vs. power optimizers. [Online]. Available: https://www.energysage.com/solar/101/string-inverters-microinverters-power-optimizers/.

A Product Comparisons for an Off-Grid System

Solar Panels

For a photovoltaic (PV) system, the PV panel is the most essential block, and is used to harvest as much energy as possible and convert it into usable electricity.

Solar panels are made out of semiconductor materials. The most common material is silicon. When light hits a semiconductor, the material absorbs a certain portion of its energy. This energy effectively knocks electrons loose, allowing them to flow freely. Solar panels also have an electric field, which forces these free electrons to flow in a certain direction, creating a unidirectional current.

Types of Solar Panels

There are many types of solar cells on the market today. For example, Monocrystalline, Polycrystalline, Thin Film, Building-Integrated Photovoltaics and Copper-Indium-Gallium-Selenide. Most commonly, manufacturers generally use one of three processes, which are Monocrystalline (also known as Mono-Si), Polycrystalline (also known as Multicrystalline or Multi-Si), and Thin Film. Figure 116 shows the market shares of each technology over the past 25 years, and Table 28 compares advantages and disadvantages of these three types of solar panels.

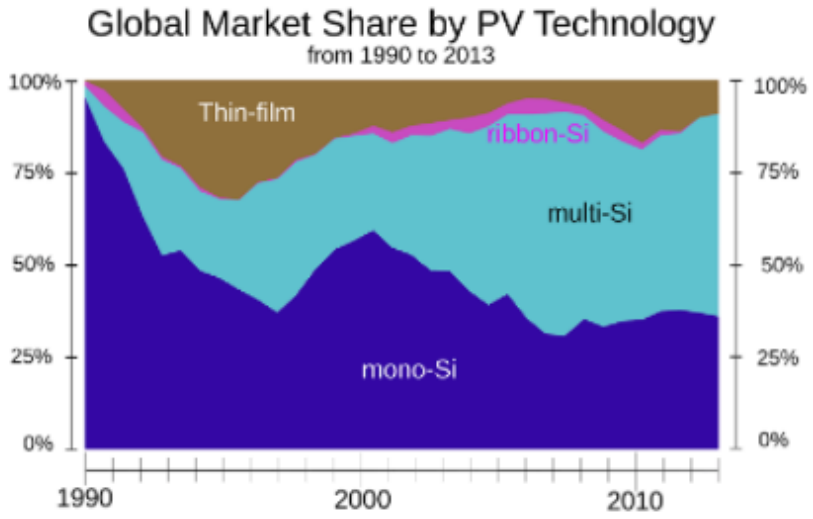


Figure 116: Market share of different types of PV panel [43]

PV Type	Advantages	Disadvantages
Mono-crystalline	• More efficient (11-22% Efficient) • Long life spans • Small size	• More expensive • Wasteful production process that produces excess silicon
Poly-crystalline	• Less expensive (No filtering process) • Less Waste	• A lower average efficiency rate (13%-16%) • Larger in size because of low efficiency • Need a filtering process
Thin Film	• Lightweight	• Very low efficiency (7%-13%) • High cost • Larger in size for similar power output because of low efficiency • Short life span

Table 28: Comparison of PV panel types [44]

Table 29 gives a comparison of several similar mono- and poly-crystalline solar panels. Thin film is not included because of its much lower efficiency, making it an unlikely choice for a low power system like ours.

Manufacturer	Mighty Max Solar	Allpowers	Mighty Max Solar	REC Solar
PV Cell Type	Monocrystalline	Monocrystalline	Polycrystalline	Polycrystalline
Output Voltage	12V	12V	12V	29.7V
Optimal Power	100W	100W	100W	240W
Working voltage [Vmp]:	18V	18V	18.2V	29.7V
Working current [Imp]:	5.60A	5.56A	5.49A	8.17A
Maximum system voltage:	1000V	1000V	1000V	-
Open-Circuit Voltage (Voc)	22.4V	20V	22.8V	34.4V
Short-Circuit Current (Isc)	5.87A	5.80A	5.95A	7.03A
Efficiency	15% - 20%	up to 23.5%	-	14.5%
Cost	$124	$162 for 100W $93 for 50W	$99	$150
Size	48 x 22 x 1.80 inches	560 x 540 x 2.5 inches	48 x 22 x 1.80 inches	65.5 x 39 x 1.5 inches
Notes	Lowest cost	-	Efficiency is not advertised	Output voltage is not 12 V

Table 29: Comparison of commercial PV panels

Battery Charge Controller

A battery charge controller's primary function is to regulate voltage from a solar panel to charge a battery with maximum efficiency. The charge controller prevents overcharging or completely draining a battery, and monitors the reverse current flow from the battery. In addition, as is described in detail in the following section, a charge controller can also help to maximize energy drawn from a solar panel. Therefore, using a charge controller for a PV system can increase the lifetime of a battery.

Types of charge controller

- Simple 1 or 2 Level Controllers: These controllers have shunt transistors to control the voltage in one or two steps. When it senses that the solar panel voltage is too high (even if the panel is producing useful power), it just shorts or disconnects the solar panel. These controllers are very low cost. However, they operate with such low efficiency that it is hard to buy one on the market today.

- PWM (Pulse Width Modulated): This is the traditional type of charge controller, and is essentially the industry standard now, especially for low power applications. PWM controllers usually have up to 80% efficiency, but are much cheaper than MPPT controllers.

- Maximum power point tracking (MPPT): The MPPT solar charge controller is the most advanced technology in today's PV systems. These controllers identify the

most productive working voltage and amperage of the solar panel given the current sun conditions. The outcome is extra 10-30% more power out of a solar array versus a PWM controller. However, MPPT controllers are much more expensive than simple 1 or 2 level controllers and PWM controllers.

Figure 117 shows the current-voltage and power-voltage relationships for a PV cell. For any given set of operating conditions, a solar panel has a unique short circuit current (I_{SC}) and open circuit voltage (V_{OC}), with its voltage varying between 0V and V_{OC} and current varying between 0A and I_{SC}. Just as the voltage nears V_{OC} and current begins to dramatically drop off towards 0A, there is a point known as the "Maximum Power Point" where the panel produces the most power possible in the given conditions. An MPPT charge controller tracks the maximum power point, and draws power from the solar panel at the voltage required to operate the solar panel at that point. It then uses DC/DC conversion to change the voltage to that which is used for charging the battery [45].

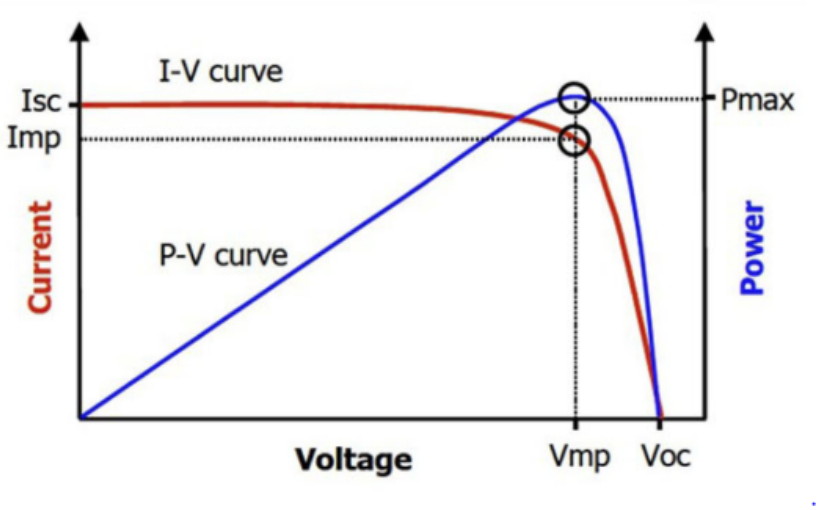


Figure 117: Maximum power point of solar panel I-V curve [46]

There are several MPPT algorithms in use, which all attempt to maximize the power from a PV panel. The most commonly implemented MPPT algorithms include, but are not limited to, the constant voltage (CV), perturb & observe (P&O), incremental conductance (INC), open-circuit voltage (VOC), and short-circuit current (ISC) methods. Table 118, which was prepared by a previous MQP report, compares those five algorithms.

Characteristics of methods	P&O	INC	VOC	CV	ISC
Choose a maximum power point using a predetermined value or ratio of values			x	x	x
Measurements of the PV open circuit voltage are taken, and an approximation for V_{MPP} is calculated as approximately 0.76-0.78 of V_{OC}			x	x	
Determines I_{MPP} as a ratio of I_{sc} to locate the maximum power point (MPP)					x
Provide dynamic MPPT under changing irradiance, temperature, and voltage conditions	x	x			

Figure 118: Comparison of five MPPT algorithms

Table 30 compares several commercially available charge controllers.

Name	Victron Energy BlueSolar Charge Controllers	SunSaver MPPT Solar Controller	Renogy Commander	Renogy Wanderer (Without MPPT)
Manufacturer	Victron Energy BlueSolar	SunSaver	Renogy	Renogy
Controller Type	MPPT	MPPT	MPPT	PWM
Solar Panel Type	12/24V nominal	12V, 24V, or 36V nominal	High voltage/string	12V
Output options (battery, DC loads)	Battery and/or DC load, but no inductive loads	Battery and/or DC load, but no inductive loads	Battery and/or DC load, but no inductive loads	Battery and/or DC load, but no inductive loads
Battery Charging Technology	Multi-stage	4 stage	4 stage	4 stage
Max PV VOC	75V	60V	150V	150V
Vout (DC)	-	7-36V	32V max	8-72V
Iout	15A	15A	20A	30A
Pout	145W	200W	260W	400W
Efficiency (peak)	98%	95%	99%	80%
Self consumption	10mA	35mA	<60mA	<10mA
Cost	$89	$243	$154	$49

Table 30: Comparison of solar charge controllers

Battery

A battery's function in an off-grid PV system is to store energy from the PV panel and provide it to the load as needed. The PV array is used to charge the battery, and the battery voltage is then fed into the inverter to supply the AC load. Historically, the only type of battery used for energy storage with solar power systems was lead acid batteries. However Lithium-ion technology is emerging as an alternative for larger scale energy storage.

Types of Battery

The two most common battery types for power storage are lead acid batteries and lithium-ion batteries. Among these two types of batteries are several more kinds of batteries which will be briefly discussed about in the section below.

Lead-acid battery technology has been around for more than 100 years [47]. Lead-acid batteries are still used in most motor vehicles and advancements in the efficiency and lifespan of these batteries are still being made. Some of the different types of lead acid batteries are flooded, sealed, AGM, and gel, which are briefly discussed below along with Lithium-ion batteries:

- The flooded Lead Acid Battery is used primarly in engine start and traction style batteries. They usually are easy to service because the user just adds water when the battery dries out. However, they must be transported and kept upright because they have the possibility of spilling [48].

- The sealed battery is a slight modification of the Flooded battery. The difference is that the user does not have access to the cell compartments and thus has to make sure that the amount of acid sustain the chemical reaction under normal use throughout the battery warranty period [48].

- AGM is the Absorbed Glass Matte construction which is a type of sealed lead acid battery with relatively high recharge and discharge efficiency [48].

- The Gel Sealed Lead Acid Battery style is similar to AGM style, but the recharge voltages on this type of cell are lower than the other styles of lead acid battery.

- Lithium-ion batteries are considered advanced battery technology. The cells in the battery can be fully charged and discharged which improves their efficiency. Lithium-ion batteries are most commonly used in cell phones and laptops [49].

Table 31 shows some of the batteries available on the market.

Manufacturer	Battery Type	Name of Product	Capacity	Voltage	Usable Capacity	Size	Price
Crown	Deep Cycle AGM	12CRV110, 110Ah 12V Battery	110Ah	12V	594 Wh	13 x 9.44 x 6.76in (60lbs)	$235
SimpliPhi	Lithium Ferro Phosphate	PHI 655 kWh Smart-Tech 12v 51.2 Ah Battery	51.2Ah	12V	655 Wh	11.25 x 5.25 x 6.25in (16.2lbs)	$1145
Crown	Flooded Lead Acid	CR220, 6V Flooded Battery	220Ah	6V	500 Wh	10.25 x 7.06 x 9.88 in (60lbs)	$130

Table 31: Comparison of commercial batteries

Inverters

Inverter Power Types

An inverter allows independent power systems to supply conventional household appliances by converting DC to AC. The two most distinct classes of solar inverters are grid-tied and off-grid inverters.

Grid-tied Inverters

Grid tied inverters are connected to the utility grid. They are used to allow a building to supplement solar power with utility power. Some key features of grid-tied inverters include:

- Inverter synchronizes to grid frequency and voltage
- Inverter must disconnect from the grid in the case of a grid fault (anti islanding)

Two examples of grid-tied inverters that are sized similarly to our inverter have been summarized in Table 32.

	Zamp ZP-300PS	Whistler XP200i
Type of Waveform	Pure	Modified
Vin (DC)	12V	11V-15.5V
Vout (AC)	120V	110V
Pout(Continuous)	300W	200W
Cost	$59	$50
Size/Weight	7.4 x 3.7 x 2.2 inch(1.52lb)	7.5 x 11 x 7.5 inch(1lb)

Table 32: Comparison of commercial inverters

Off-Grid Inverters

Off-grid inverters cannot be connected to a system that is grid tied. The system is thus dependent entirely on the solar panels, and requires batteries to provide power when the sun is not producing sufficient power.

Residential Solar PV System and Inverter Types

There are three main types of residential inverters: microinverters, string inverters and power optimizers. These inverters can be grid-tied or off-grid, but most commercial products are grid-tied.

String Inverters

String Inverters are the oldest inverter system type and have historically been the most cost effective. Efficiencies can range up to 98% [50]. The solar panels are connected to each other in "strings" of series connected panels, which boosts the voltage into the inverter. Each of these strings is then connected to one inverter which converts DC to AC. A solar panel is essentially a current source, so a string of solar panels will only produce as much current (and thus power) as its least productive panel – if one or more of the solar panels is shaded during any part of the day, the power output from that entire string would be reduced to the level of that panel [51]. This layout is shown in Figure 119.

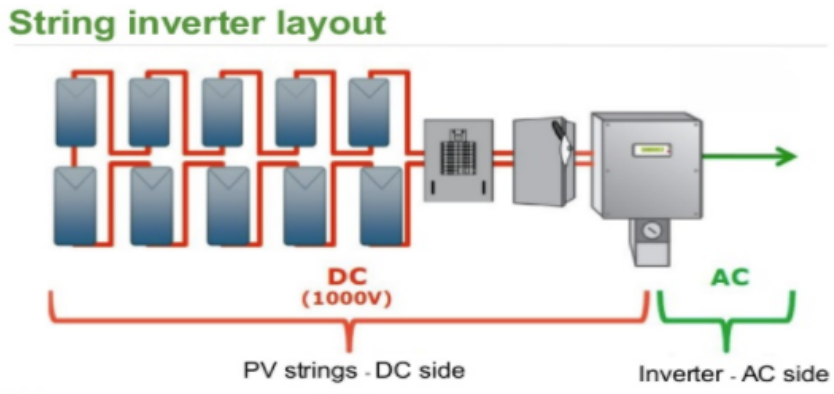


Figure 119: String inverter layout [51]

Microinverters

Microinverter efficiency is typically in the range of 90-95%. With a microinverter system, a single inverter is mounted with each individual PV panel. They convert the DC electricity from solar panels into AC electricity on the roof, with no need for a centralized string inverter. In many cases the micro-inverters are integrated into the solar panel itself, but they may also be mounted next to the panel on the mounting system [51]. This layout is shown in Figure 120.

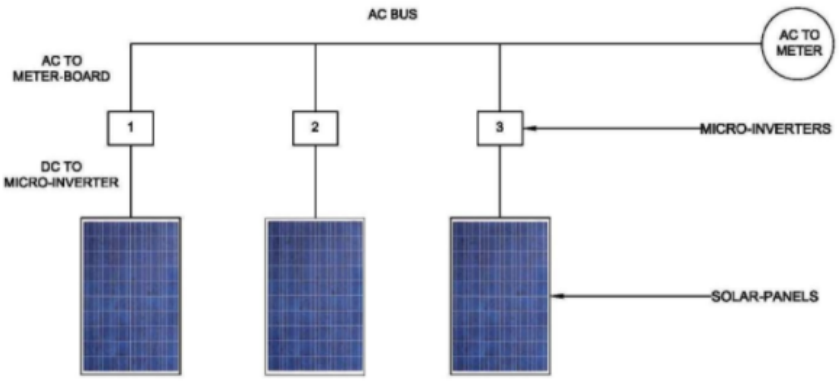


Figure 120: Microinverter layout [51]

Power Optimizers

Power Optimizers are priced between the more expensive microinverters and less expensive string inverters. Like micro-inverters, power optimizers are located at each panel, usually integrated into the panels themselves. However, instead of converting the DC electricity to AC electricity at the solar panels, they "condition" the DC electricity (through a DC/DC converter) and send it to a string inverter. The panels are in parallel, and thus can conduct different currents without shading issues. This approach results in higher system efficiency than a string inverter alone [51] This layout is shown in Figure 121.

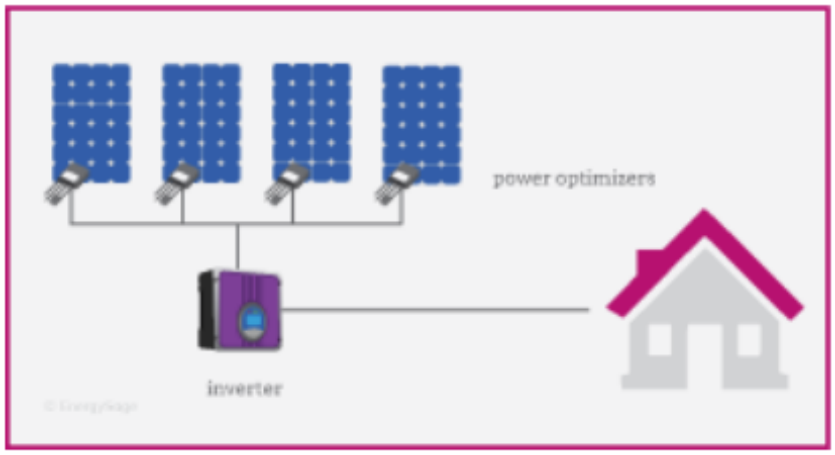


Figure 121: Power optimizer layout [51]

B Feedback System Simulations in Multisim

This section details early simulations of the inverter feedback system. We used Multisim for these simulations so that we could directly integrate the feedback system with our H-Bridge circuit. These simulations are not included in the body of the report because they ultimately turned out to be unrealistic, as will be explained. They were a good learning tool for the team, because no team members had any experience with control systems.

Ideal Circuit without H-bridge

The chosen approach for feedback simulation was to use a PID controller with once-per-cycle DC feedback provided by a peak follower. The initial simulation method for this circuit was to replace the H-Bridge with a simple sine wave generator to remove extra factor of high frequency switching. The circuit used for this simulation is shown in Figure 123:

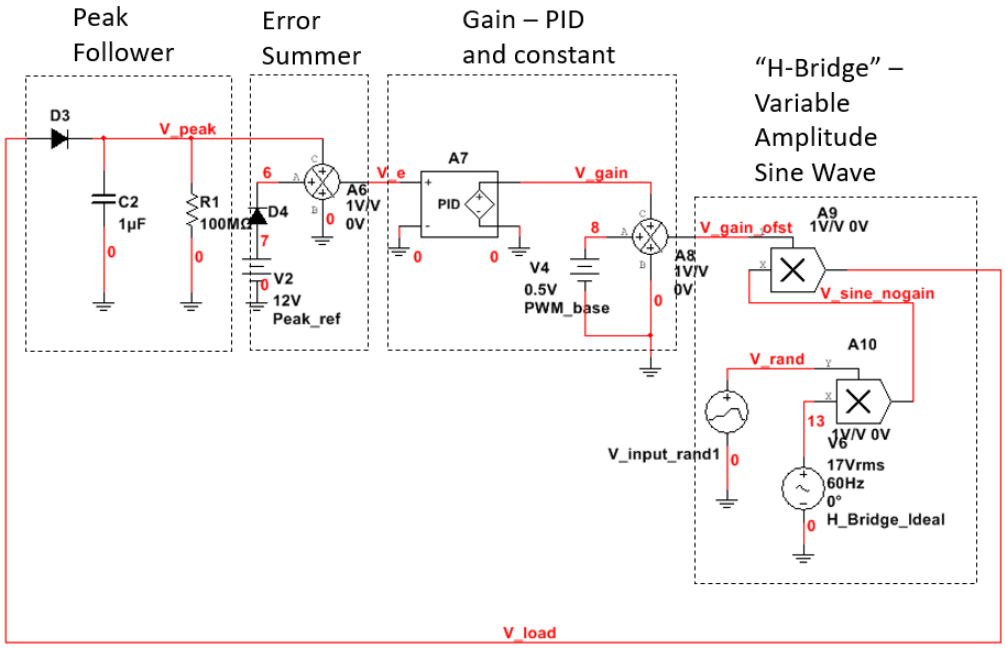


Figure 122: "Ideal" feedback test circuit for determining general feasibility of method from simulation

In this circuit, the peak follower is as previously described - the positive peaks of the sine wave are captured on the capacitor C2, which discharges through R1 so that it can capture the next sine wave peak. The error summer calculates an error signal V_e between the peak value and a setpoint - here, the setpoint is represented by a $12V_{DC}$ source fed through a diode. It is fed through a diode so that it will have an identical drop to that caused by the peak follower. The summer (which will be implemented digitally in our final project) is an ideal component that draws no current.

The next stage is the PID gain stage. Here, we have used a PID block (built into Multisim), with adjustable K_P, K_I, and K_D. As previously mentioned, K_P, K_I, and K_D are the proportional, integral, and derivative gains of the PID, respectively. Like the summer, the PID does not draw any current. We have used another summer to

add another constant gain to the PID gain, where the PID gain is defined by the PID equation, which is repeated again below:

$$v(t) = K_P * e(t) + K_i * \int_0^t e(\tau) * d\tau + K_d * de/dt \tag{17}$$

The reasoning for this is that, in this circuit, the total gain is meant to represent the amplitude modulation ratio m_a of the H-Bridge, and should vary primarily between 0V and 1V. Thus, a 0.5V offset is used so that under error-free conditions, m_a will be 0.5, a fairly reasonable value. This allows for the feedback system to function without steady state error, even when only a proportional gain is used. The need for this constant term can be eliminated by correctly using the integral term of the PID as well, as will be explored in further iterations.

To generate a sine wave in H-Bridge-like fashion, a sine wave of $17V_{RMS}$ is generated (which is 2 times the RMS voltage of a $12V_{pk}$ sine wave). This wave is multiplied by 2 terms: a random input used to represent battery or load variations, and the PID gain. Thus, with an error value (i.e. PID gain) of 0 and the constant value of 0.5, this sine wave will have an RMS value of 8.5V, and will maintain a steady state. As the random input is varied (it is a piecewise linear function), the feedback loop will respond by changing the gain value.

With this initial feedback circuit using the calculated peak detector values for R (21MΩ) and C (100pF), which are different than shown in the figure above, the output in Figure 123 was generated, where the green line is the peak follower voltage and the red is the load voltage (sine wave output voltage). After zooming in, it is quickly evident that the peak follower is too fast - it follows the load voltage through the entire positive portion of the cycle - i.e. it traces the load voltage past the peak, and then clips at 0.7V. Because of this, the comparator recognizes a very large gain, and proportionally boosts the negative half of the cycle to over 2.5kV.

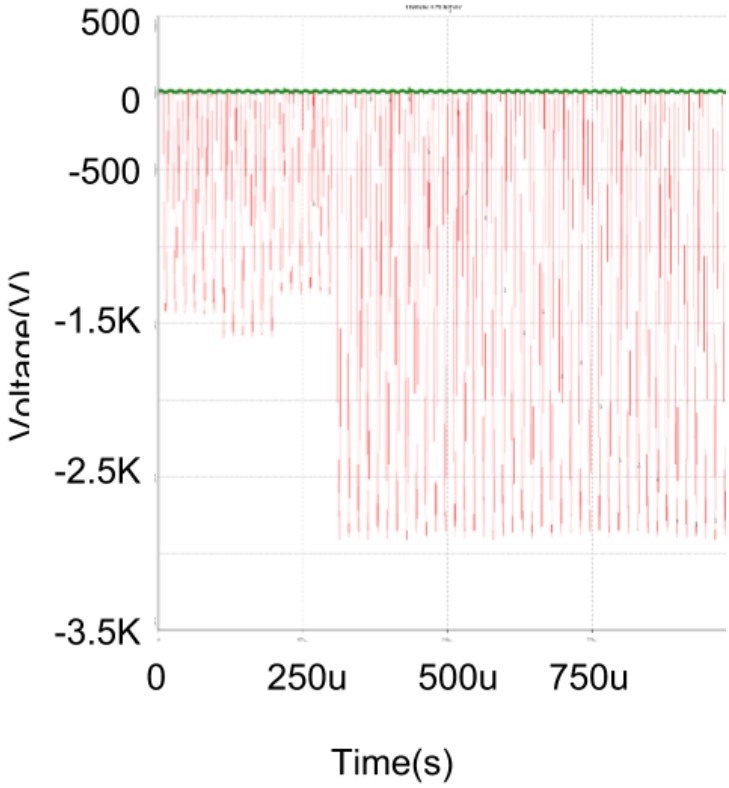


Figure 123: Peak follower (green) too fast - load voltage (red) has large negative swings from simulation

After adjusting the peak follower values to account for this unexpected behavior, an interesting note about the behavior of this circuit is noted: As the time constant increases, the quality of the sine wave and ability of the circuit to hold a constant output voltage on both the positive and negative peaks is improved. However, as time constant increases, the behavior of the circuit also becomes slower. For example, Figure 124 is an output with changing input voltage (signified by the blue line, of which the input voltage is a multiple) and a time constant of 10s. It is clear that the sine wave is often triangular or flat at its peak, and that the negative peaks are not well controlled (they increase when input voltage increases). However, the response to a large change in input voltage ($V_{\text{input rand}}$ drops from 3V to 0.5) is relatively quick, taking only 5 cycles to reach a steady output. It is worth noting that a step change this large is not expected given the slow changing nature of our battery input.

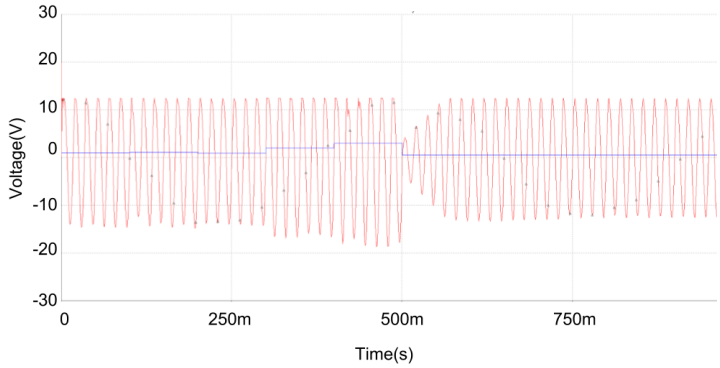


Figure 124: Tau = 10s, response (red) to changing input voltage (blue) from simulation

Figure 125: "Sine wave" peaks for the above analysis with tau = 10s from simulation

With the same input variations and $\tau = 100$s, the output in Figure 126 is generated. Here, the response is unacceptably slow, although the negative peaks tend to stay closer to -12V than in the previous test.

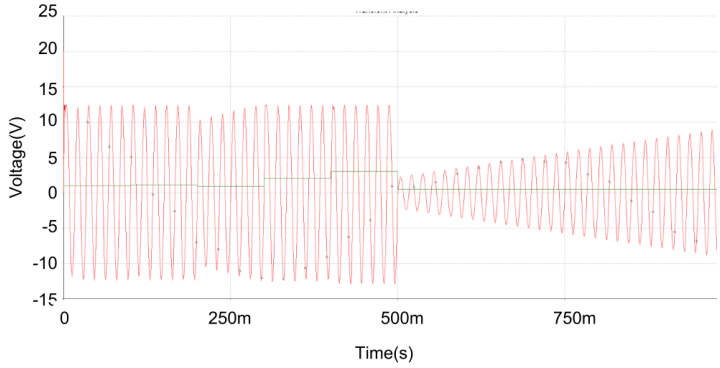


Figure 126: Tau = 100s, response (red) to changing input voltage (green) from simulation

Circuit with H-bridge

Following the simulation of the "ideal" circuit, we tested the feedback system using our ideal H-Bridge. This circuit is shown in Figure 127.

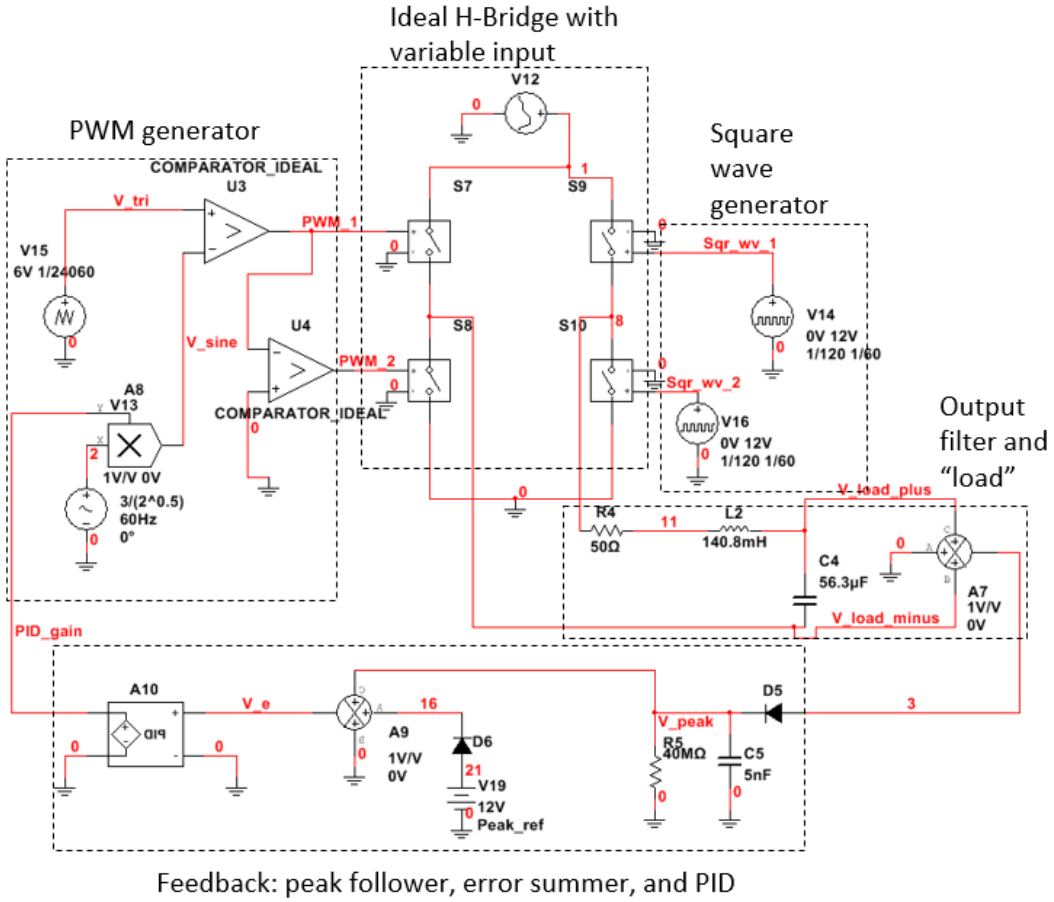


Figure 127: Ideal H-Bridge feedback simulation circuit from simulation

This circuit is a combination of our H-Bridge circuit (explained previously) and the same feedback system used previously. One key difference in both circuits is that the PID gain now directly multiplies the amplitude of the sine wave used to generate PWM (in the PWM generator block). With this version of the circuit, we will attempt to tune the PID controller, so the DC offset added to the PID gain is removed. The load feedback signal is generated by taking the difference between the positive and negative sides of the load. We have generated several initial simulations with this circuit. In the following simulations the time constant τ= 100 seconds, with R = 40MΩ and C = 5µF. All of these circuits use a constant input voltage of 12V while the circuit is tuned. For the initial simulation, the PID terms were chosen as: $K_P = 0.25$, $K_I = 25$, $K_D = 0$ (i.e. derivative term is not used - this term will be considered once stable operation is achieved):

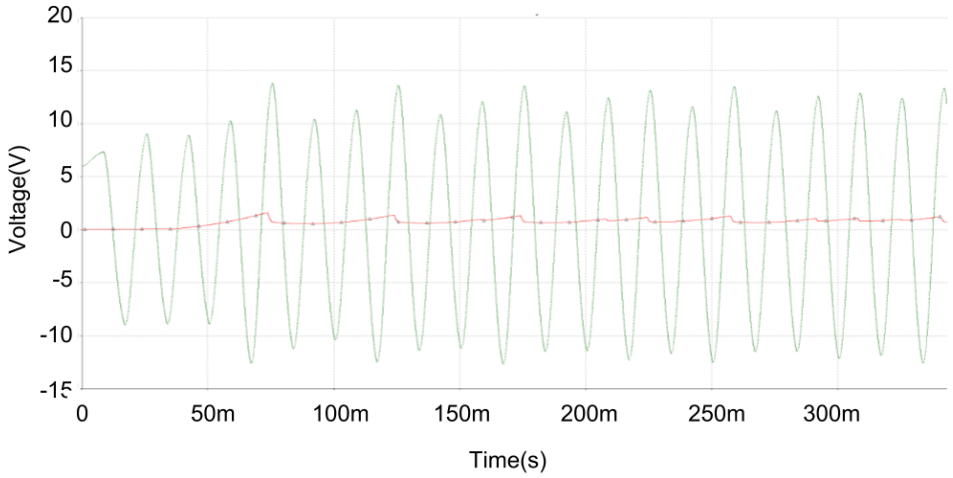


Figure 128: Feedback simulation with ideal H-Bridge, $K_P = 0.25$, $K_I = 25$, $K_D = 0$. The green sine wave is the load voltage, and the red line is the PID gain.

It is clear that the above system is unstable - the peaks of the load voltage oscillate repeatedly, when the system should be at steady state. It is noticeable that on the taller positive peaks of the sine wave (i.e. those that exceed the set-point of 12V), a relatively significant negative change in the PID gain occurs, due to the proportional gain. This explains the corresponding drop in peak amplitude on the following sine wave cycle. As the integral gain grows, the sine wave grows and again exceeds 12V - so the oscillations repeat. Thus, the next step is to reduce the proportional gain, so that it provides a less severe response when the voltage exceeds 12V. The simulation in Figure 129 has the values $K_P = 0.1$, $K_I = 25$, $K_D = 0$. It can be seen that there are still some oscillations, but they are noticeably smaller. This performance should be further improved to provide a truly stable operation, but the variations in this voltage may be within the bounds of our 115VAC$\pm$10% bounds (after the transformer). More measurements will need to be completed to determine if these oscillations are acceptable after being stepped up through the transformer.

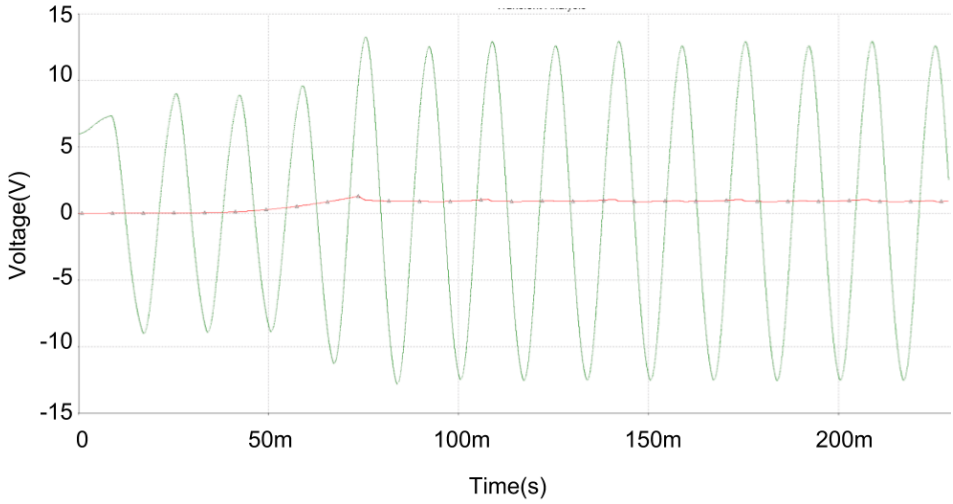


Figure 129: Feedback simulation with ideal H-Bridge, $K_P = 0.1$, $K_I = 25$, $K_D = 0$. The green sine wave is the load voltage, and the red line is the PID gain.

It was ultimately decided that we should abandon these simulations because of their dissimilarity with our discrete time, microcontroller based feedback on our actual inverter product. However, this was a useful exploration into the functioning of a PID and feedback loop.

C Microcontroller Code

C.1 Header file: MQP_PWM.h

```
/*
* MQP_PWM.h
*
* Created on:  Nov 15, 2017
* Author:  Ryan Cooney
*/

#ifndef MQP_PWM_H_
define MQP_PWM_H_

// system clock frequency
#define SIXTYMHZ 60000000

// for 60Hz PWM
// these rely on 60MHz CPU frequency and PWM clock prescaled by 16 to 3.75MHz
#define PWM_60HZ_TIMER_MAX 31250
#define PWM_60HZ_TIMER_HALF 15625

// for high frequency sine wave PWM
#define PI 3.14159265358979323846
#define SWITCH_FREQ 2400

// for ADC/ePWM3 and feedback
#define SAMPLING_FREQ 24000
#define SAMPLE_BUF_SIZE 1024 // this is more than enough space at 48kHz...  increase
size if frequency increases
#define BUFFER_WRAP(i) (i) & (SAMPLE_BUF_SIZE - 1)
#define ADC_HALF 1900 // as of 1-24-18, this is the "center" of sine wave from
feedback circuit
//2039 // approx.  1.65V, based off of measurements
#define ADC_FULL 4095 // 3.3V
#define AVG_BUF_SIZE 50 // number of samples to maintain for rolling average
#define AMP_DELTA 3 // measured in ADC readings

// PID
#define VOLTAGE_SETPOINT 590 // measured in ADC buckets - V = 3.3 * ADC_READING/2Î2
#define INTEGRAL_MAX (1L << 22)

// ADC state machine
typedef enum {
ADC_INIT = 0,
ADC_RUN
} ADC_STATE;
```

```
// a few example deadband time constants - 60Hz
// min possible deadtime with 60MHz system clock and 16x prescale is about 260ns
#define DEADBAND_60Hz_MIN 1

// note that GLOBAL_Q is defined in IQmathLib.h
// not defined here
// GLOBAL_Q will need to change as switching frequency changes, which is important!


#endif /* MQP_PWM_H_ */
```

C.2 C file: MQP_PWM.c

```
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
//// ////
//// MQP_PWM.c ////
//// Generate 3-level PWM and Control Amplitude ////
//// Ryan Cooney ////
//// 2-19-2018 ////
//// With initialization code taken from Example_F2802xEpwmDeadBand.c,////
//// from TI controlSUITE ////
//// ////
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////
// PROGRAM DESCRIPTION
//////////////////////////////////////////////////////////////////////////
//
// GOALS:
// 1.  Generate high frequency PWM with modulated duty cycle to generate sign
wave,
// output complementary PWM signals on ePWM1a and ePWM1b
// 2.  Generate 60Hz PWM with 50% duty cycle and output complementary
// signals on ePWM2a and ePWM2b
// 3.  Integrate dead time into both signals
// 4.  Setup ADC0 to sample feedback sine wave
// 5.  Use I/Q sampling to calculate amplitude of sine wave
// 6.  Use PID to calculate error and adjust PWM modulation ratio
//
//////////////////////////////////////////////////////////////////////////


/*
*
* Includes
*
*/
#include "DSP28x_Project.h" // Device Headerfile and Examples Include File
#include "f2802x_common/include/adc.h"
#include "f2802x_common/include/clk.h"
#include "f2802x_common/include/flash.h"
#include "f2802x_common/include/gpio.h"
#include "f2802x_common/include/pie.h"
#include "f2802x_common/include/pll.h"
#include "f2802x_common/include/pwm.h"
#include "f2802x_common/include/timer.h"
#include "f2802x_common/include/wdog.h"
#include "IQmathLib.h"
#include "MQP_PWM.h" // my header

/*
*
* Function prototypes
```

```
*
*/
void InitEPwm1(void);
void InitEPwm2(void);
void InitEPwm3(void);
void InitADC1(void);
void InitGPIO(void);
void InitTimer1(void);

__interrupt void epwm1_isr(void);
__interrupt void epwm2_isr(void);
__interrupt void epwm3_isr(void);
__interrupt void adc_isr(void);

uint32_t cpu_load_count(void);


/*
*
* Globals
*
*/
// for accessing "objects"
ADC_Handle myADC;
CLK_Handle myClk;
FLASH_Handle myFlash;
GPIO_Handle myGpio;
PIE_Handle myPie;
PWM_Handle myPwm1, myPwm2, myPwm3;
TIMER_Handle myTimer;


// for sine wave PWM generation
const uint32_t switchFreqDiv60Min1 = (SWITCH_FREQ / 60) - 1;
const double phaseMultiplier = 60 * 2 * PI / SWITCH_FREQ;
_iq phaseMultIQ;
const uint64_t timerPeriodL = (SIXTYMHZ / (2*(uint64_t)SWITCH_FREQ));
const uint64_t timerPeriodHalfL = (SIXTYMHZ / (4*(uint64_t)SWITCH_FREQ));
const uint16_t timerPeriod = (uint16_t)(timerPeriodL);
const uint16_t timerPeriodHalf = (uint16_t)(timerPeriodHalfL);
long timerHalfLong = (long)timerPeriodHalf;


// for taking sine of phase, updated in ISR
long EPwm1TimerIntCount;
_iq phaseOut;
_iq EPwm1CompareIQ;
long EPwm1CompareValLong;
long EPwm1CompareValLong1;
volatile uint16_t EPwm1CompareVal16;
_iq28 EPwm1CompareValIQ28;


// counter used to help ISR decide when to sync the two PWM modules
int16_t syncCount = 0;
int nextSync = 0;
```

```
// for measuring CPU load
uint32_t count_unloaded, count_loaded;
volatile float cpu_load; // volatile so it isn't optimized away - for now only
being used in debugger


// for getting current sine and cosine values
uint16_t samplesPer90 = (SAMPLING_FREQ >> 3) / 60;
uint16_t sampleBuf[SAMPLE_BUF_SIZE]; // store 1/4 cycle worth of previous samples,
to simulate cosine wave...  but we make buf twice as big as this to be safe
uint16_t bufIndex = 0; // index of most recent sample
uint16_t cosNew = 0; // current value of cosine, which is just sine with 90 degree
lag from real sine wave
int16_t sinOffset;
int16_t cosOffset;
_iq sinIQ;
_iq cosIQ;


// for calculating amplitude
long sineAmp;
_iq ampAvg;
long ampAvgInt = 0;
_iq oldAmpScale;
_iq newAmpScale;

// for PID
_iq dt = _IQ((double)1/SAMPLING_FREQ);
_iq integral = 0;
_iq derivative = 0;
_iq err = 0;
_iq errPrev = 0;
_iq gainPID = 0;
_iq28 gainIQ28;
_iq Kp; // constant
_iq Ki; // constant
_iq Kd; // constant


// ADC state machine
ADC_STATE ADCstate = ADC_INIT;


// A misguided effort at turning feedback off until output is stable
_iq ampAvgAvg;
long ampAvgAvgInt;
int oscillatingAmpCount = 1500;


/*
*
* Extern globals
*
*/
// for copying .econst from FLASH to RAM at runtime
extern unsigned int econst_loadstart;
extern unsigned int econst_loadsize;
extern unsigned int econst_runstart;
```

```
// for copying ISRs from FLASH to RAM at runtime
extern unsigned int RAM_ISRs_loadstart;
extern unsigned int RAM_ISRs_loadsize;
extern unsigned int RAM_ISRs_runstart;

// for copying IQmath to RAM
extern unsigned int IQmath_loadstart;
extern unsigned int IQmath_loadsize;
extern unsigned int IQmath_runstart;


/*
*
* Macros
*
*/
// Maximum Dead Band values
#define EPWM1_MAX_DB 0x03FF
#define EPWM2_MAX_DB 0x03FF
#define EPWM3_MAX_DB 0x03FF

#define EPWM1_MIN_DB 0
#define EPWM2_MIN_DB 0
#define EPWM3_MIN_DB 0



/*
*
*
* main()
*
*
*/
void main(void) {
CPU_Handle myCpu;
PLL_Handle myPll;
WDOG_Handle myWDog;

// copy constants (the .econst section) to RAM from flash
memcpy(&econst_runstart, &econst_loadstart, (Uint32)&econst_loadsize);
memcpy(&RAM_ISRs_runstart, &RAM_ISRs_loadstart, (Uint32)&RAM_ISRs_loadsize);
memcpy(&IQmath_runstart, &IQmath_loadstart, (Uint32)&IQmath_loadsize);

// Initialize all the "object" handles
myADC = ADC_init((void *)ADC_BASE_ADDR, sizeof(ADC_Obj));
myClk = CLK_init((void *)CLK_BASE_ADDR, sizeof(CLK_Obj));
myCpu = CPU_init((void *)NULL, sizeof(CPU_Obj));
myFlash = FLASH_init((void *)FLASH_BASE_ADDR, sizeof(FLASH_Obj));
myGpio = GPIO_init((void *)GPIO_BASE_ADDR, sizeof(GPIO_Obj));
myPie = PIE_init((void *)PIE_BASE_ADDR, sizeof(PIE_Obj));
myPll = PLL_init((void *)PLL_BASE_ADDR, sizeof(PLL_Obj));
myPwm1 = PWM_init((void *)PWM_ePWM1_BASE_ADDR, sizeof(PWM_Obj));
myPwm2 = PWM_init((void *)PWM_ePWM2_BASE_ADDR, sizeof(PWM_Obj));
myPwm3 = PWM_init((void *)PWM_ePWM3_BASE_ADDR, sizeof(PWM_Obj));
myTimer = TIMER_init((void *)TIMER0_BASE_ADDR, sizeof(TIMER_Obj));
myWDog = WDOG_init((void *)WDOG_BASE_ADDR, sizeof(WDOG_Obj));
```

```
// Perform basic system initialization
WDOG_disable(myWDog);
CLK_enableAdcClock(myClk);
(*Device_cal)(); // pointer to a function call that calibrates ADC and oscillators
CLK_disableAdcClock(myClk);

// Select the internal oscillator 1 as the clock source
CLK_setOscSrc(myClk, CLK_OscSrc_Internal);

// Setup the PLL for clock x 12 / 2 which will yield 60Mhz = 10Mhz * 12 / 2
PLL_setup(myPll, PLL_Multiplier_12, PLL_DivideSelect_ClkIn_by_2);

// Disable the PIE and all interrupts
PIE_disable(myPie);
PIE_disableAllInts(myPie);
CPU_disableGlobalInts(myCpu);
CPU_clearIntFlags(myCpu);

// If running from flash copy RAM only functions to RAM
#ifdef _FLASH
memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);
#endif

// Setup a debug vector table and enable the PIE
PIE_setDebugIntVectorTable(myPie);
PIE_enable(myPie);

// Register interrupt handlers in the PIE vector table
PIE_registerPieIntHandler(myPie, PIE_GroupNumber_3, PIE_SubGroupNumber_1,
(intVec_t)&epwm1_isr);
PIE_registerPieIntHandler(myPie, PIE_GroupNumber_10, PIE_SubGroupNumber_1,
(intVec_t)&adc_isr);

// init GPIO for PWM outputs
InitGPIO();

// init ADC for feedback sampling
InitADC1();

// init ePWM
CLK_disableTbClockSync(myClk);

InitEPwm1();
InitEPwm2();
InitEPwm3();

CLK_enableTbClockSync(myClk);

// Initialize counters:
EPwm1TimerIntCount = 0;

// initialize timer for CPU load
InitTimer1();

// enable interrupts for ADC and ePWM
CPU_enableInt(myCpu, CPU_IntNumber_3); // CPU INT3 is connected to EPWM1-3 INT
CPU_enableInt(myCpu, CPU_IntNumber_10); // CPU INT10 = ADC1
```

```
PIE_enablePwmInt(myPie, PWM_Number_1); // EPWM channel 1 interrupt
PIE_enableAdcInt(myPie, ADC_IntNumber_1); // ADC channel 1 interrupt

// initial count for CPU load
count_unloaded = cpu_load_count();

// initialize variables
phaseMultIQ = _IQ28(phaseMultiplier);
float radPerSampleF = 60*2*PI / (SAMPLING_FREQ >> 1); // divide samp_freq by 2
b/c we currently have a 2-step conversion sequence?  so interrupts only being
triggered half as often as they should be

gainPID = _IQ(1.0); // just an initial condition, this will change as PID runs
Kp = _IQ(0.01); // constant gain multipliers
Ki = _IQ(0.03);
Kd = _IQ(0);

float newAmpScaleFloat = 0.002;
newAmpScale = _IQ(newAmpScaleFloat);
oldAmpScale = _IQ(1 - newAmpScaleFloat);
ampAvg = _IQ(VOLTAGE_SETPOINT);


ampAvgAvg = _IQ(VOLTAGE_SETPOINT);



// Enable global Interrupts and higher priority real-time debug events
CPU_enableGlobalInts(myCpu);
CPU_enableDebugInt(myCpu);

/*
*
* Main loop
*
*/
//volatile int myInt = 0;
for(;;)
{
// measure CPU load
count_loaded = cpu_load_count();
cpu_load = 1.0f - (float)count_loaded/count_unloaded; // compute CPU load

// __asm(" NOP"); // uncomment if not measuring CPU load
}
}



//////////////////////////////////////////////
//
//
//
// INTERRUPT SERVICE ROUTINES
//
//
//
//////////////////////////////////////////////
```

```
/*!
* Updates compare value for ePWM1 (high speed PWM)
*/
#pragma CODE_SECTION(epwm1_isr, "RAM_ISRs") // indicate to linker that this function
is its own section, which will be run from RAM
__interrupt void epwm1_isr(void)
{

// get sine wave compare val for adjusting PWM duty cycle
phaseOut = _IQ28mpyI32(phaseMultIQ, EPwm1TimerIntCount);
EPwm1CompareIQ = _IQ28sin(phaseOut); // take sine and store as _iq

EPwm1CompareValLong = _IQ28mpyI32int(EPwm1CompareIQ, (long)timerPeriodHalf);
// scale by height of ePWM triangle but save as a SIGNED long because IQ value
has sign

// To integrate feedback, multiply compare value by gain
EPwm1CompareValIQ28 = _IQ28mpy(EPwm1CompareIQ, (long)timerPeriodHalf);
gainIQ28 = _IQtoIQ28(gainPID);
EPwm1CompareValLong1 = _IQ28mpyI32int(EPwm1CompareValLong, gainIQ28);

if (EPwm1CompareValLong < (-1*timerHalfLong)) {
EPwm1CompareValLong = (-1*timerHalfLong);
}

if (EPwm1CompareValLong1 < (-1*timerHalfLong)) {
EPwm1CompareValLong1 = (-1*timerHalfLong);
}

EPwm1CompareVal16 = (uint16_t)(EPwm1CompareValLong1 + (timerPeriodHalf));

PWM_setCmpA(myPwm1, EPwm1CompareVal16);
//PWM_setCmpA(myPwm2, EPwm1CompareVal16); // for two level PWM


// increment sine value
if (EPwm1TimerIntCount == switchFreqDiv60Min1) {
EPwm1TimerIntCount = 0;
} else {
EPwm1TimerIntCount++;
}

if (nextSync == 1) {
nextSync = 0;
PWM_forceSync(myPwm1);
}

// after 1 60Hz cycle, synchronize low and high frequency
if (EPwm1CompareIQ == _IQ28(-1)) {
// force sync on next go
nextSync = 1;
}

// Clear INT flag for this PWM channel
PWM_clearIntFlag(myPwm1);
```

```
// Acknowledge this interrupt to receive more interrupts from group 3
PIE_clearInt(myPie, PIE_GroupNumber_3);
}


/*
* ISR that...
* 1.  Is triggered by EOC from the ADC, and saves the ADC value
* 2.  Finds the phase of the sine wave based on ADC input
* 3.  Performs DQ transform to get magnitude of sine wave
* 4.  Implements PID to give feedback gain
*
*
*/
#pragma CODE_SECTION(adc_isr, "RAM_ISRs") // indicate to linker that this function
is its own section, which will be run from RAM
__interrupt void adc_isr(void)
{
// update buffer index before we store new data
bufIndex = BUFFER_WRAP(bufIndex + 1);

// read result and store in circular buffer
sampleBuf[bufIndex] = ADC_readResult(myADC, ADC_ResultNumber_1);

// grab 90 degree phase shifted value of sine wave (i.e.  cosine that is in phase
with sine)
// samplesPer90 needs to be almost perfect for this to have a 90 degree lag
cosNew = 2*ADC_HALF - sampleBuf[BUFFER_WRAP(bufIndex - samplesPer90)];

/*
* calculate amplitude of sine wave...
*/
// sin^2(theta) + cos^2(theta) = 1
// a^2 * sin^2(theta) + a^2 * cos^2(theta) = a^2
// sqrt(a^2 * sin^2(theta) + a^2 * cos^2(theta)) = a^2
sinOffset = sampleBuf[bufIndex] - ADC_HALF;
cosOffset = cosNew - ADC_HALF;
sinIQ = (long)sinOffset * 128L; // same as but faster than:  _IQ7(sinOffset);
cosIQ = (long)cosOffset * 128L; // same as but faster than:  _IQ7(cosOffset);
sineAmp = _IQ7int(_IQ7mag(sinIQ, cosIQ)); // IQ function takes sqrt(a^2 + b^2)

/*
* Low pass filter
*/
ampAvg = _IQmpy(ampAvg, oldAmpScale) + _IQmpy(_IQ(sineAmp), newAmpScale);
ampAvgInt = _IQint(ampAvg);

// check difference between current amplitude reading and average
// if X samples in a row have a difference of less than Y,
// we've *probably* reached a steady state and can turn feedback on
ampAvgAvg = _IQmpy(ampAvgAvg, _IQ(0.8)) + _IQmpy(ampAvg, _IQ(0.2));
ampAvgAvgInt = _IQint(ampAvgAvg);

if ((ampAvgAvgInt - ampAvgInt) > 10 || (ampAvgInt - ampAvgAvgInt) > 10 /*|| ampAvgInt
> 700*/ || ampAvgInt < 480) {
oscillatingAmpCount += 1;
if (oscillatingAmpCount > 1500) {
oscillatingAmpCount = 1500;
```

```
}
} else {
oscillatingAmpCount -= 15;
if (oscillatingAmpCount < 0) {
oscillatingAmpCount = 0;
}
}


// hysteresis state changer
if (ADCstate == ADC_INIT && oscillatingAmpCount < 250) {
ADCstate = ADC_RUN;
} else if (ADCstate == ADC_RUN && oscillatingAmpCount > 1400) {
ADCstate = ADC_INIT;
}


/*
* STATE 0:  INIT
*/
if (ADCstate == ADC_INIT) {
gainPID = _IQ(1.0);
}

/*
* STATE 1:  RUN
*/
else {

/*
*
* PID
* with basic code from here:  https://softwareengineering.stackexchange.com/questions/18
*
*/
err = _IQ(VOLTAGE_SETPOINT - ampAvgInt);

// take integral
integral = integral + _IQmpy(err, dt);
if (integral > INTEGRAL_MAX) {
integral = INTEGRAL_MAX;
} else if (integral < -1*INTEGRAL_MAX) {
integral = -1*INTEGRAL_MAX;
}

// take derivative
derivative = (err - errPrev) / dt;
errPrev = err;

// get gain
gainPID = _IQmpy(Kp, err) + _IQmpy(Ki, integral) + _IQmpy(Kd, derivative);// +
_IQ(1);

}

ADC_clearIntFlag(myADC, ADC_IntNumber_1);
PIE_clearInt(myPie, PIE_GroupNumber_10);
```

```
return;
}


////////////////////////////////////////////////
//
//
//
// INITIALIZATION FUNCTIONS
//
//
//
////////////////////////////////////////////////

/*
* Sinusoidal PWM
*/
void InitEPwm1()
{
// myClk is 60MHz as of 11-16
CLK_enablePwmClock(myClk, PWM_Number_1);

// start with a low frequency, say 2kHz
// frequency is calculated by...
// T_PWM = 2*TBPRD/TBCLK
// F_PWM = 1/T_PWM
// where...
// TBPRD is user selected period (in clock ticks)
// TBCLK is prescaled frequency of SYSCLKOUT
// f_PWM is pwm frequency
PWM_setPeriod(myPwm1, timerPeriod); // Set timer period
PWM_setPhase(myPwm1, 0x0000); // Phase is 0
PWM_setCount(myPwm1, 0x0000); // Clear counter

// Setup TBCLK
PWM_setCounterMode(myPwm1, PWM_CounterMode_UpDown); // Count up and down
PWM_disableCounterLoad(myPwm1); // Disable phase loading
PWM_setHighSpeedClkDiv(myPwm1, PWM_HspClkDiv_by_1); // Clock ratio to SYSCLKOUT
PWM_setClkDiv(myPwm1, PWM_ClkDiv_by_1); // Slow just to observe on the scope

// Setup compare
PWM_setCmpA(myPwm1, timerPeriodHalf);

// Set actions
PWM_setActionQual_CntUp_CmpA_PwmA(myPwm1, PWM_ActionQual_Set);
PWM_setActionQual_CntDown_CmpA_PwmA(myPwm1, PWM_ActionQual_Clear);

PWM_setActionQual_CntUp_CmpA_PwmB(myPwm1, PWM_ActionQual_Clear);
PWM_setActionQual_CntDown_CmpA_PwmB(myPwm1, PWM_ActionQual_Set);

// Active High complementary PWMs - setup the deadband
PWM_setDeadBandOutputMode(myPwm1, PWM_DeadBandOutputMode_EPWMxA_Rising_EPWMxB_Falling);
PWM_setDeadBandPolarity(myPwm1, PWM_DeadBandPolarity_EPWMxB_Inverted);
PWM_setDeadBandInputMode(myPwm1, PWM_DeadBandInputMode_EPWMxA_Rising_and_Falling);
PWM_setDeadBandRisingEdgeDelay(myPwm1, 200*10);//EPWM2_MIN_DB);
PWM_setDeadBandFallingEdgeDelay(myPwm1, 200*10);//EPWM2_MIN_DB);
```

```
// Interrupt where we will modify the compare point
PWM_setIntMode(myPwm1, PWM_IntMode_CounterEqualZero); // Select INT on Zero event
PWM_enableInt(myPwm1); // Enable INT
PWM_setIntPeriod(myPwm1, PWM_IntPeriod_FirstEvent); // Generate INT on 3rd event

}


/*
* 60Hz PWM
*/
void InitEPwm2()
{
CLK_enablePwmClock(myClk, PWM_Number_2);

// frequency is calculated by...
// T_PWM = 2*TBPRD/TBCLK
// F_PWM = 1/T_PWM
// where...
// TBPRD is user selected period (in clock ticks)
// TBCLK is prescaled frequency of SYSCLKOUT
// f_PWM is pwm frequency
PWM_setPeriod(myPwm2, PWM_60HZ_TIMER_MAX); // Set timer period - measured in clock
ticks, so depends on prescaled clock freq of 60MHz
//PWM_setPeriod(myPwm2, 6250);
PWM_setPhase(myPwm2, 0x0000); // Phase is 0
PWM_setCount(myPwm2, 0x0000); // Clear counter

// Setup TBCLK
PWM_setCounterMode(myPwm2, PWM_CounterMode_UpDown); // Count up
//PWM_disableCounterLoad(myPwm2); // Disable phase loading
// Actually, we ENABLE phase loading - this allows PWM1 to force PWM2 to synchronize
PWM_enableCounterLoad(myPwm2);
PWM_setHighSpeedClkDiv(myPwm2, PWM_HspClkDiv_by_4); // Clock ratio to SYSCLKOUT
PWM_setClkDiv(myPwm2, PWM_ClkDiv_by_4);

// Load registers when only when count hits 0, to ensure proper synchronization
PWM_setShadowMode_CmpA(myPwm2, PWM_ShadowMode_Shadow); // Load registers every
ZERO
PWM_setShadowMode_CmpB(myPwm2, PWM_ShadowMode_Shadow);
PWM_setLoadMode_CmpA(myPwm2, PWM_LoadMode_Zero);
PWM_setLoadMode_CmpB(myPwm2, PWM_LoadMode_Zero);

// Setup compare
PWM_setCmpA(myPwm2, PWM_60HZ_TIMER_HALF);
//PWM_setCmpA(myPwm2, 3125);

// Set actions
// PWM1A goes high when we hit compare value on the way up, and low when we hit
it on the way down
// PWM1B is the opposite, it goes low when we hit compare value on the way down
and high on the way up
PWM_setActionQual_CntUp_CmpA_PwmA(myPwm2, PWM_ActionQual_Set);
PWM_setActionQual_CntDown_CmpA_PwmA(myPwm2, PWM_ActionQual_Clear);

PWM_setActionQual_CntUp_CmpA_PwmB(myPwm2, PWM_ActionQual_Clear);
PWM_setActionQual_CntDown_CmpA_PwmB(myPwm2, PWM_ActionQual_Set);
```

```
// Active high complementary PWMs - Setup the deadband
// deadband time calculation is the same as frequency calculation
PWM_setDeadBandOutputMode(myPwm2, PWM_DeadBandOutputMode_EPWMxA_Rising_EPWMxB_Falling);
PWM_setDeadBandPolarity(myPwm2, PWM_DeadBandPolarity_EPWMxB_Inverted);
PWM_setDeadBandInputMode(myPwm2, PWM_DeadBandInputMode_EPWMxA_Rising_and_Falling);
PWM_setDeadBandRisingEdgeDelay(myPwm2, 100*DEADBAND_60Hz_MIN);
PWM_setDeadBandFallingEdgeDelay(myPwm2, 100*DEADBAND_60Hz_MIN);

}



/*
* FOR 2 LEVEL PWM USE THIS VERSION
*/
//void InitEPwm2()
//{
// // myClk is 60MHz as of 11-16
// CLK_enablePwmClock(myClk, PWM_Number_2);
//
// // start with a low frequency, say 2kHz
// // frequency is calculated by...
// // T_PWM = 2*TBPRD/TBCLK
// // F_PWM = 1/T_PWM
// // where...
// // TBPRD is user selected period (in clock ticks)
// // TBCLK is prescaled frequency of SYSCLKOUT
// // f_PWM is pwm frequency
// PWM_setPeriod(myPwm2, timerPeriod); // Set timer period
// PWM_setPhase(myPwm2, 0x0000); // Phase is 0
// PWM_setCount(myPwm2, 0x0000); // Clear counter
//
// // Setup TBCLK
// PWM_setCounterMode(myPwm2, PWM_CounterMode_UpDown); // Count up and down
// PWM_disableCounterLoad(myPwm2); // Disable phase loading
// PWM_setHighSpeedClkDiv(myPwm2, PWM_HspClkDiv_by_1);//PWM_HspClkDiv_by_1); //
Clock ratio to SYSCLKOUT
// PWM_setClkDiv(myPwm2, PWM_ClkDiv_by_1);//PWM_ClkDiv_by_1); // Slow just to observe
on the scope
//
// // Setup compare
// PWM_setCmpA(myPwm2, timerPeriodHalf);
//
// // Set actions
// PWM_setActionQual_CntUp_CmpA_PwmA(myPwm2, PWM_ActionQual_Clear);
// PWM_setActionQual_CntDown_CmpA_PwmA(myPwm2, PWM_ActionQual_Set);
//
// PWM_setActionQual_CntUp_CmpA_PwmB(myPwm2, PWM_ActionQual_Set);
// PWM_setActionQual_CntDown_CmpA_PwmB(myPwm2, PWM_ActionQual_Clear);
//
// // Active High complementary PWMs - setup the deadband
// PWM_setDeadBandOutputMode(myPwm2, PWM_DeadBandOutputMode_EPWMxA_Rising_EPWMxB_Falling);
// PWM_setDeadBandPolarity(myPwm2, PWM_DeadBandPolarity_EPWMxB_Inverted);
// PWM_setDeadBandInputMode(myPwm2, PWM_DeadBandInputMode_EPWMxA_Rising_and_Falling);
// PWM_setDeadBandRisingEdgeDelay(myPwm2, 200*10);//EPWM2_MIN_DB);
// PWM_setDeadBandFallingEdgeDelay(myPwm2, 200*10);//EPWM2_MIN_DB);
//
// // Interrupt where we will modify the compare point
```

```
// PWM_setIntMode(myPwm2, PWM_IntMode_CounterEqualZero); // Select INT on Zero
event
// PWM_enableInt(myPwm2); // Enable INT
// PWM_setIntPeriod(myPwm2, PWM_IntPeriod_FirstEvent); // Generate INT on 3rd
event
//
//}



/*
* Init ePWM3, which is actually used to trigger sampling for the ADC
*/
void InitEPwm3(void) {

// determine period based on frequency specified in header file
// PWM is in up-count mode, so T_PWM = (TBPRD + 1) * T_TBCLK
// at 60MHz, T_TBCLK = 16.67ns

uint64_t timer3PeriodL = ((SIXTYMHZ)/ ((uint64_t)SAMPLING_FREQ)) - 1;
uint16_t timer3Period = (uint16_t)(timer3PeriodL);
uint16_t timer3PeriodHalf = (uint16_t)(timer3PeriodL / 2);

CLK_enablePwmClock(myClk, PWM_Number_3);

// send Start of Conversion (SOC) pulse while counting up, when CmpA value is
hit
PWM_enableSocAPulse(myPwm3);
PWM_setSocAPulseSrc(myPwm3, PWM_SocPulseSrc_CounterEqualCmpAIncr);
PWM_setSocAPeriod(myPwm3, PWM_SocPeriod_FirstEvent);
((PWM_Obj *)myPwm3)->CMPA = 0;
PWM_setPeriod(myPwm3, timer3Period);
PWM_setCounterMode(myPwm3, PWM_CounterMode_Up);
}


/*
* Init ADC1 for reading feedback
*/
void InitADC1(void) {

CLK_enableAdcClock(myClk);

ADC_enableBandGap(myADC);
ADC_enableRefBuffers(myADC);
ADC_powerUp(myADC);
ADC_enable(myADC);
ADC_setVoltRefSrc(myADC, ADC_VoltageRefSrc_Int);

ADC_setIntPulseGenMode(myADC, ADC_IntPulseGenMode_Prior);
ADC_enableInt(myADC, ADC_IntNumber_1); // there's no IntNumber_0, so use IntNumber_1
ADC_setIntMode(myADC, ADC_IntNumber_1, ADC_IntMode_ClearFlag); // a new interrupt
is not generated until interrupt flag is cleared
ADC_setIntSrc(myADC, ADC_IntNumber_1, ADC_IntSrc_EOC1);
ADC_setSocChanNumber (myADC, ADC_SocNumber_0, ADC_SocChanNumber_A0); // Pin 26
on launchpad
ADC_setSocChanNumber (myADC, ADC_SocNumber_1, ADC_SocChanNumber_A0);
ADC_setSocTrigSrc(myADC, ADC_SocNumber_0, ADC_SocTrigSrc_EPWM3_ADCSOCA);
```

```
ADC_setSocTrigSrc(myADC, ADC_SocNumber_1, ADC_SocTrigSrc_EPWM3_ADCSOCA);
ADC_setSocSampleWindow(myADC, ADC_SocNumber_0, ADC_SocSampleWindow_37_cycles);
// Sets sample and hold time for 37 cycles
ADC_setSocSampleWindow(myADC, ADC_SocNumber_1, ADC_SocSampleWindow_37_cycles);

}



/*
* Any GPIO pins that are used should be enabled here
* Currently have pins enabled for PWM1a,1b,2a,2b
*/
void InitGPIO(void) {
// ePWM1 - 60Hz square wave
GPIO_setPullUp(myGpio, GPIO_Number_0, GPIO_PullUp_Disable);
GPIO_setPullUp(myGpio, GPIO_Number_1, GPIO_PullUp_Disable);
GPIO_setMode(myGpio, GPIO_Number_0, GPIO_0_Mode_EPWM1A);
GPIO_setMode(myGpio, GPIO_Number_1, GPIO_1_Mode_EPWM1B);

// ePWM2 - High speed sine wave PWM
GPIO_setPullUp(myGpio, GPIO_Number_2, GPIO_PullUp_Disable);
GPIO_setPullUp(myGpio, GPIO_Number_3, GPIO_PullUp_Disable);
GPIO_setMode(myGpio, GPIO_Number_2, GPIO_2_Mode_EPWM2A);
GPIO_setMode(myGpio, GPIO_Number_3, GPIO_3_Mode_EPWM2B);
}


/*
* Init Timer1 to a 10ms timer, to be used for the CPU load count
* Don't start the timer, we'll use it as a "one-shot"
*/
void InitTimer1(void) {
TIMER_stop(myTimer);
TIMER_setPeriod(myTimer, 600000); // 10ms at 6MHz is 600,000 clock ticks
TIMER_setPreScaler(myTimer, 0);
TIMER_reload(myTimer);
TIMER_setEmulationMode(myTimer, TIMER_EmulationMode_StopAfterNextDecrement);
}


/*
* Use this function to take base CPU load estimate while interrupts are disabled,
then update value every 10ms after interrupts are enabled
* Counts amount of timer ticks in 10ms - if it is preempted by interrupts, it
misses ticks
* Can estimate CPU load based on how many ticks it misses
*
*/
uint32_t cpu_load_count(void)
{
uint32_t i = 0;

// Start "one-shot" timer
TIMER_reload(myTimer);
TIMER_start(myTimer);
while (TIMER_getStatus(myTimer) == TIMER_Status_CntIsNotZero) {
i++;
```

```
}
TIMER_stop(myTimer);
return i;
}
```